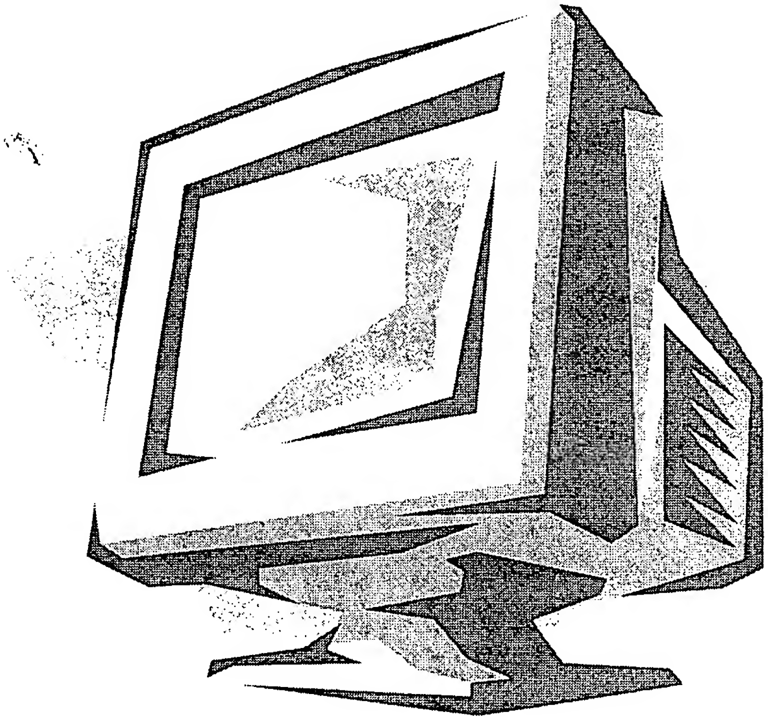


WebSpective LA

Installation and
User's Guide

Atreve Software, Incorporated

Information described in this manual is furnished for information only, is subject to change without notice, and should not be construed as a commitment by ASI. ASI assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual.

Printing History

Date	**Document**	**Release**
07/16/97	01-000000-100-0042-b02	WebSpective 1.0 Beta 2 Installation and User's Guide
08/19/97	01-000000-100-0053	WebSpective 1.0 Installation and User's Guide

Printed in U.S.A.

Table of Contents

About This Guide

Navigating the Manual i
Style Conventions i
Common Terms ii
Interface Terms ii
Web Site Terms ii
Feedback iii

Product Overview

What is WebSpective? 1
High Availability 1
Traffic Control and Scalability 1
General Manageability 2
WebSpective Architecture 2
The Interceptor 2
The Manager 3
The Agent 3
The Filter 4
The Viewer 4
System Requirements 4
Hardware and Operating System 4
File System Requirements 5
Security Considerations 5

Installation and Configuration

Installation Basics 6
Network Installations 6
Third-Party Requirements 6
Making Connections 7
Multi-homing Web Servers 7
Windows NT Installation 9
The First Component 9
Subsequent Components 9
UNIX Installation 11
Local Installation 11
Setting Up Files with pkgadd 11
Making a Keyfile 12
Setting Environment Variables 12
SSL Support 13
Filter Installation 13
Installation on a Netscape FastTrack or Enterprise Server 14
How to edit the "obj.conf" file 15
Installation on a Microsoft IIS server 16

How to install the Filter on the Microsoft IIS server 16
Testing the Filter installation. 16
Configuration **17**
How to run the configuration utility 17

Operating WebSpective

Start-Up and Local Control **19**
How to start a WebSpective component 19
Shutdown 20
The Viewer **21**
Navigating with the Mouse 21
The Management Tab **22**
Exploring the Web Site 22
Creating Custom Views 22
How to create a new view 22
How to edit a custom view 23
How to save a view under a new name 23
How to delete a view 23
The Creation Bar 23
The Health Tab **23**
Preconfiguration for Windows NT-based Agents 23
How to enable disk and network performance monitoring 23
View, Panel, and Graph Creation 24
How to create or edit a view. 24
How to create or edit a panel 24
How to create or edit a graph. 25
The Event Tab **26**

Site Management

Object States **28**
Object Properties. **30**
How to change a property using "Set All Values". 31
Property Definitions 31
Logging Fields 38
How to select which fields are logged 39
Object Management **40**
How to add objects to the WebSpective system 40
Deactivating and Reactivating Objects. 41
How to deactivate an object. 41
How to reactivate an object 42
Stopping and Restarting Objects 42
How to stop an object. 42
How to restart an object. 42
Deleting Objects. 43
How to delete objects. 43

Appendix A: Registry Parameters

Agent Registry. A-1
Application Registry. A-2
Filter Registry A-4
Interceptor Registry. A-5

Manager Registry A-8
Viewer Registry A-9

Appendix B: Database Structur s

Database Architecture B-1
The Hit Logging Table B-3
Metric Logging Tables B-4
Interfaces Table B-6
Events Table B-7

Appendix C: System Behavior

Logged Events. C-1
Client Connections C-1

Appendix D: Security Considerations

The Keyfile. D-1
Component Communication D-1
Components and setuid (UNIX systems only). D-1
Filesystem security D-2

Appendix E: Firewall Considerations

Glossary

Index

About This Guide

The WebSpective User's Guide is intended to help you quickly learn to make productive use of WebSpective functionality. In order to aid in this process, this guide uses a number of conventions that are exercised throughout the book. These conventions are described in this chapter, along with a list of tasks and topics with which a WebSpective User should be familiar.

Chapter Topics

- Navigating the Manual
- Style Conventions
- Common Terms

Navigating the Manual

In order to help you move more quickly through the manual, a table of contents and index are provided in addition to a topic listing that occurs at the beginning of each chapter. Topics in the topic listing appear in the order in which they occur in the chapter.

Style Conventions

The following style conventions have been instituted to help you discern different kinds of information in the book:

```
System text is used wherever a command-line entry is required
```

How-to headings

How-to headings signify the beginning of a step-by-step operation. They are listed in italics in the table of contents to help differentiate them from regular sub-headings.

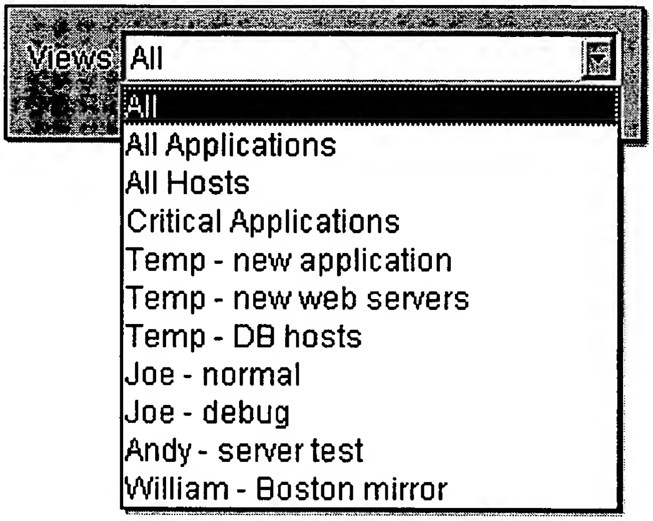
- **Bolded Items**—are items which are key to WebSpective's operation. Bolded items are usually configuration parameters or menu options.

Common T rms

The following terms are used frequently throughout the guide. Make sure that you understand how these terms are used in this guide as their definitions may differ from more standard usage.

Interface Terms

Combo Box—An interface element which contains a dialog box and a pull-down menu as follows:



Click—Push the left mouse button over the specified object on the screen.

Select—Choose from a menu or set of options by clicking on the menu or set and dragging the mouse pointer down to the desired object. Select can also be used interchangeably with Click.

Web Site Terms

Daemon—A kind of process which runs for a long period of time in the "background" (meaning that there is no direct control interface). While a daemon is running, it waits for specific conditions to occur and then performs an action.

Endpoint—As used by Atreve Software, Inc., an endpoint is an IP address and port number which locate a specific connection to a web server. In the WebSpective management model, each of these endpoints exists to provide the content of a single Application.

Host—The physical machine on which processes are run.

IP Interface—An interface on a host where a set of instructions (the *Internet Protocol*) allows processes on the host to interact with processes on other hosts.

Port—The destination (represented by a number) on a given IP interface for a connection between a remote process and a process "listening" at the interface.

Process—An instance of a program running on a machine. In UNIX, processes can be listed by typing "ps" at the command line. In Windows NT, processes can be listed by calling the task manager (by pressing and holding <CTRL><ALT><DEL> and then selecting the Task Manager from the menu that pops up).

URL—*Uniform Resource Locator*, an address string that identifies a document or resource on the World Wide Web. A URL is a pointer either to a file on the local machine, or to a file elsewhere on the Internet.

Web Server—A kind of daemon which waits for requests from a client and then returns files which contain HTML. In this guide, the terms "web server" and the more generic "server" are synonymous.

Feedback

We want to hear from you! If you have questions or comments about the WebSpective User's Guide, you can contact the Atreve Software Documentation Team in one of the following ways:

Fax:	(617) 354-0513
E-Mail:	documentation@atreve.com
Postal Mail:	

Documentation Team c/o
Atreve Software, Inc.
767C Concord Avenue
Cambridge, MA. 02138

Note that there is also a survey supplied with this User's Guide that you can use to send comments to the Documentation Team.

Chapter 1

Product Overview

Atreve Software, Inc. proudly introduces its high volume web management tool, WebSpective. Designed to optimize sites which average ten thousand or more hits per day, WebSpective turns the task of running multiple servers (with varied content) into an easy, single-user operation.

Chapter Topics:

- What is WebSpective?
- WebSpective Architecture
- System Requirements
- Security Considerations

What is WebSpective?

In simplest terms, WebSpective is a site management solution for business-critical, high volume and transactional web sites. The software's overall operation is divided between three different focus areas:

High Availability

By maintaining a continuous and dynamic watch over a web site's vital signs, WebSpective can deliver unprecedented fault tolerance and recovery. WebSpective can attempt to correct faults on web servers and can start overflow servers if they are available to the web site. WebSpective's own design defends against software faults and can be configured with redundant components to further ensure crash resistance.

Traffic Control and Scalability

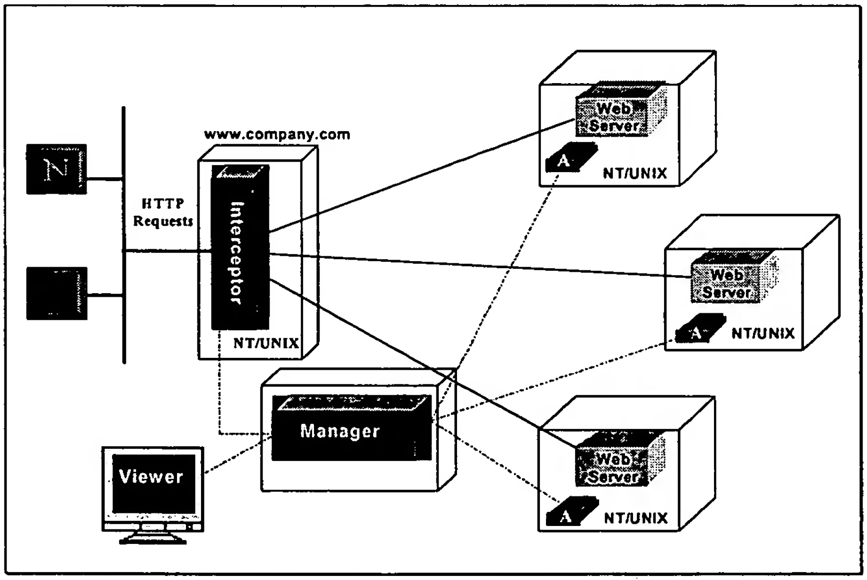
WebSpective's first line of hit management is called Interceptor. Interceptor runs both as part of WebSpective or as a stand-alone module. Interceptor takes incoming hits and distributes them across on and off-site web servers. On its own, Interceptor can also be used to control access to web content. Combined with the rest of the WebSpective package, Interceptor's effectiveness increases with dynamic performance updates from the website. WebSpective continuously manages hit distribution patterns to optimize the load across web servers.

General Manageability

The other side of WebSpective's operation is a host of administration tools that other web management programs cannot provide. WebSpective keeps a database of vital information about a site's servers and can be configured to take action in response to a number of hit- and system-related events. Comprehensive, dynamic server and content profiles are provided along with access to fundamental site tools in a single user interface.

WebSpective Architecture

The WebSpective system is comprised of five elements: the Interceptor, the Agent, the Filter, the Manager, and the Viewer.



The Interceptor

The purpose of the Interceptor is to route requests for a particular application to a server available to handle that application, and to balance the load of requests among those servers. It receives updates of current server load and of administrative changes from the Manager; this includes changes in server availability (starts and stops) and site-wide start and shutdown actions.

The Interceptor's primary design goals are turnaround speed to clients making HTTP requests, and security against attack, as it is necessarily the most exposed component of WebSpective, and perhaps the most critical.

The Manager

The Manager is the central "brain" of WebSpective. Though the Interceptor and Agents will operate correctly in the absence of a Manager, all dynamic configuration and functionality beyond a basic framework is directed through it. Specifically:

- The Manager forwards system information from Agents to the database and the user (via the Viewer).
- The Manager directs Agents to start and stop processes
- The Manager updates the Interceptor with dynamic load information for its routing, and with changes in system configuration (adding or removing servers).
- The Manager handles alerts from Agents and the Interceptor, reacting as necessary. For example, an alert from an Agent causes the Manager to direct the Interceptor to avoid affected endpoints until further notice.

Even if the Manager fails, the system will run indefinitely. However, the Interceptor performance becomes sub-optimal, the Agents begin to discard log information, and the Viewer is unable to connect to the WebSpective system.

The Agent

The Agent serves as the intermediary between the Manager and all of the web servers on a given host. The Agent has the task of periodically reporting hit data and performance statistics to the Manager, which come from these sources:

- The Agent receives hit data from the Filter, which is inside the web server
- The Agent periodically samples the host machine for its performance statistics
- The Agent periodically executes a "ping" of the web server, to guarantee that it is functional from a client-side view, and to measure its response time and TCP queue.

The Agent is also responsible for processing requests from the Manager that affect the Agent itself and the web servers it controls.

If a server goes down unexpectedly, the Agent attempts to restart it. The Agent logs the restart, and continues to attempt to restart the server until a specified limit has been reached. When a server goes down the Interceptor is immediately notified by Manager. Hits are steered away by the Interceptor until the Agent successfully restarts the server.

The Filter

The Filter brings the web server into the WebSpective system, recording hit data and allowing the Agent (at the Manager's request) to start, stop, and control the server. Currently, Netscape and Microsoft IIS servers are the only ones supported by WebSpective; however, other servers will be added in later releases.

The Filter consists of a shared library or DLL which conforms to either the Netscape API (NSAPI) or the Microsoft Internet Server API (ISAPI). It is invoked from the server via the NSAPI or ISAPI, and acts as a conduit between the Agent and the server. If, for example, the Agent has determined that the server is overloaded and some requests should be redirected, it instructs the Filter to do this.

The Viewer

The Viewer allows you to control every aspect of the WebSpective system. By connecting the user with the Manager, the viewer transforms the WebSpective system into a highly configurable web management tool. The Viewer enables the user to make highly customized views of desired elements of the website. The Viewer also provides users with dynamic graphs with which to monitor the performance of web site elements.

System Requirements

The following system requirements must be met in order for WebSpective to run properly.

Hardware and Operating System

Any WebSpective component will run on Intel-compatible systems running Microsoft's Windows NT 4.0, and on sparc-compatible systems running SunSoft's Solaris 2.5.1

TABLE 1. **WebSpective Hardware and OS Requirements by Component**

Component	**Minimum RAM (Mb)**	**CPU Time**
Interceptor	16	Most to All (should be on a dedicated machine)
Manager	32	Most (can share processor with the user's database)
Filter and Agent	32	Minimal (runs with multiple web servers on a single host)
Viewer	16	Minimal

The Databas . WebSpective does not require the use of a user-supplied database engine, but can provide additional functionality for sites which incorporate one. The engine may be any supported by the Java Database Connection, such as Oracle. Additionally, that list includes any engine supporting the ODBC protocol such as Microsoft SQL server.

The Web Servers. For WebSpective 1.0, the web servers must be Netscape (Commerce, FastTrack 2.0, or Enterprise 2.0) or Microsoft IIS 3.0 servers

File System Requirements

All components require access to syslog (on UNIX systems) or the Event-Viewer (on Windows NT). All components have an identical secret keyfile that is used for authentication of information from other components. On UNIX systems, each component must be able to access a configuration file which they must read on startup

The Manager and Agents keep a database (The Managed Object Database, or MODB) of the entities that they manage which is read and written to during operation. Also, depending on the customer's chosen database and JDBC driver, additional file system access for read and/or write may be needed to log hits on the web site.

Security Considerations

WebSpective has been designed as a very secure system, robust to attacks and misinformation. Refer to Appendix D: Security Considerations for a full discussion of security and the WebSpective system.

Chapter 2

Installation and Configuration

Chapter Topics:

- Installation Basics
- Windows NT Installation
- UNIX Installation
- Filter Installation
- Configuration

Installation Basics

On UNIX systems, installation begins with the user running the pkgadd utility. On Windows NT systems, installation is managed by the InstallShield program.

Network Installations

It is important to note that for a given component of the WebSpective system, all pertinent files should be installed locally. This means that files should not be installed on network drives, and should also not be installed on NFS mounted drives. Some files cannot be accessed properly if they are not on the local machine, and the practice of local installation further enhances overall site security.

Third-Party Requirements

The performance of the WebSpective system relies in part upon third-party products, such as Java. Consequently, it is important that you acquire and install certain programs before you can properly install WebSPective. Refer to the Support area of the Atreve web site at: "http://www.atreve.com/html/support.html" for the current list of third-party requirements.

Making Connections

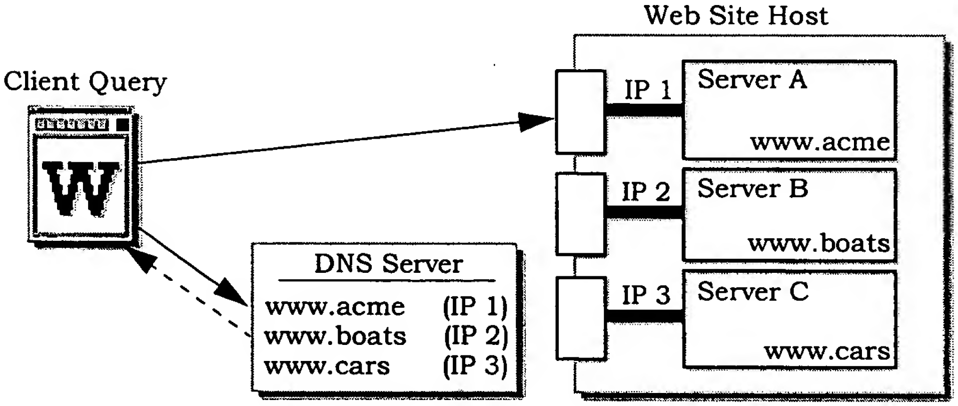
As you add WebSpective components, you should be ready to specify host and port settings for components you have already installed. Specifically, the Manager will need to know where the Interceptor is located, and the Agents and Interceptor will in turn need to know the location of the Manager. New components use this information to "register" themselves with earlier components and to open channels of communication with them.

Multi-homing Web Servers

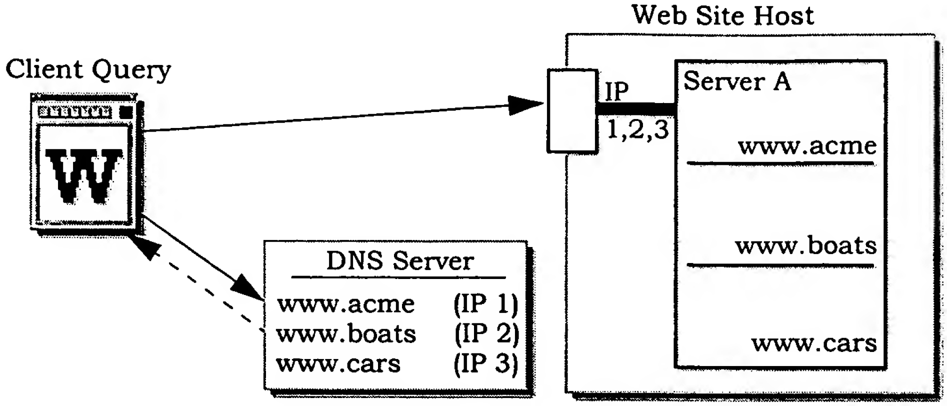
Many sites use a combination of hardware and software technology to produce a multi-homing environment. This allows a single machine with a single network card to host multiple web sites or applications.

Multi-homing Architectures. These virtual servers can be grouped into three general categories: multi-process, hardware virtual, and software virtual.

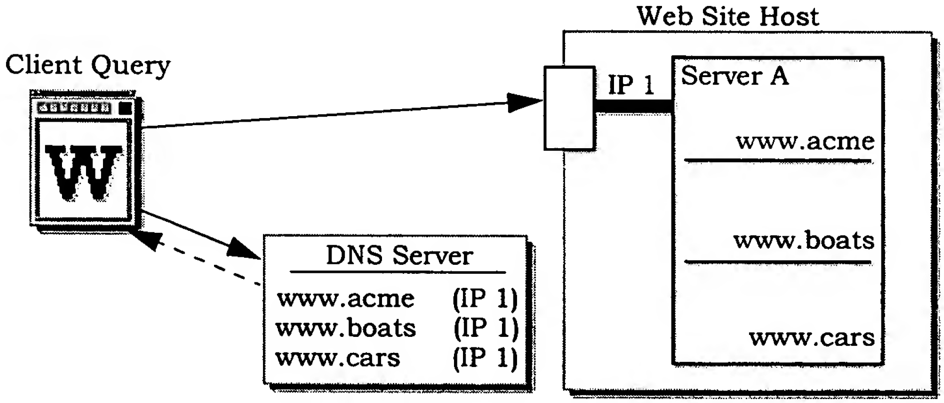
- **Multi-Process**: In multi-process distribution, several web servers run on the same host machine, each with their own IP address. Each server can provide a different application.



- **Hardwar Virtual**: Unlike in multi-process distribution, only one server runs on the host machine. The network card passes requested IP addresses to the server, indicating what content should be sent out to the browser.



- **Software Virtual**: In software virtual distribution the web server reads the headers of a hit request to determine what virtual server the client wants. This method only works if the client browser request to the server includes the desired virtual server name. Some older browsers do not provide this information.



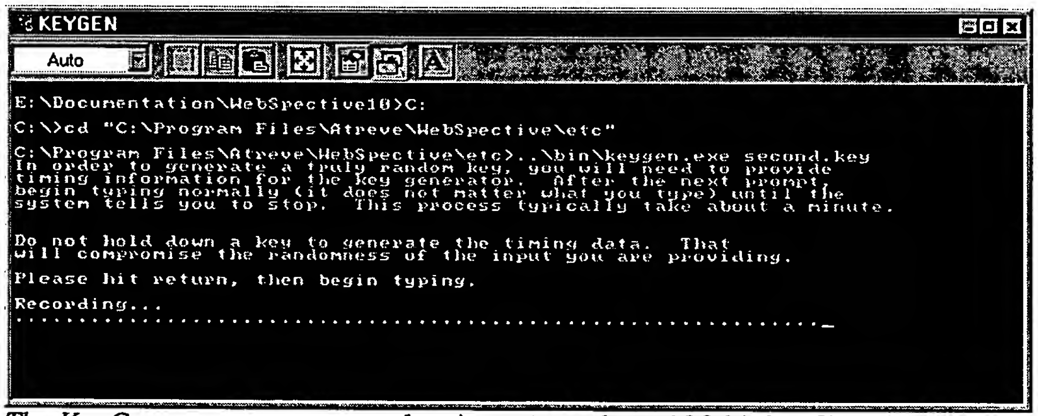
WebSpective and the Interceptor support multi-process and hardware virtual servers. They do not currently support software-virtual distribution.

Windows NT Installation

On Windows NT systems, installation of the various components is done with InstallShield. To begin installation, locate and run the "Setup.exe" file for the component you wish to install. While you can install the components in any order, Atreve recommends that you start by installing the Manager on its host machine.

The First Component

When you install the first component of your WebSpective system, InstallSheild will generate an Installation Parameters file and a keyfile (described in Appendix D: Security Considerations). The Installation Parameters file contains common information that is used by all of the components of the system and is only necessary during component installation. You should create this file on a floppy disk or a network drive so that you can access it as you install other components. The keyfile generation requires a randomly timed string of user-entered characters as directed by the install tool. You will also be prompted to copy this to the floppy disk, ensuring that each component will get an identical copy of the keyfile.



```
E:\Documentation\WebSpective10>C:

C:\>cd "C:\Program Files\Atreve\WebSpective\etc"

C:\Program Files\Atreve\WebSpective\etc>..\bin\keygen.exe second.key
In order to generate a truly random key, you will need to provide
timing information for the key generator.  After the next prompt,
begin typing normally (it does not matter what you type) until the
system tells you to stop.  This process typically take about a minute.

Do not hold down a key to generate the timing data.  That
will compromise the randomness of the input you are providing.

Please hit return, then begin typing.

Recording...
...............................................................
```

The KeyGen program uses random input to make a 128-bit keyfile

Subsequent Components

All components require a name and a location on the local file system. Additionally, all components will want to read the Installation Parameters file and keyfile that you created during the installation of the first component. Finally, all components will allow you to customize their IP and Port addresses as well as those of the Manager. These values can also be modified later—see "Configuration" on page 17 for more information.

lter. You need to install a Filter into every web server that you intend to use with the WebSpective system. Refer to page 13 for Filter installation information.

Databas s. The Manager and the Agent both create internal databases that contain information on the objects that they manage (an MODB). During the installation of these components, you will be asked to specify a name and location for their database, which is stored as a pair of files. The Manager can also be configured to access an external database that is used to log historical data. The particulars of this logging database (LDB) are described in Appendix B. During the Manager's installation you will be able to specify a data source for this database, a user name for the Manager, and a password.

The LDB is not necessary to WebSpective's operation. However, access to this external database will allow you to compile and interpret a great deal of information from your web site. You must have an LDB to do the following:

- Use the Health Tab, which takes information from the LDB to provide real-time and historical performance graphs
- Centralize information by recording server log files and event histories to one location where they can be cataloged and read through
- Keep track of machine statistics to determine hardware performance and usage over time data for your website

Configuration. In addition to putting files in their proper place, the installation tool sets configuration parameters that determine the behavior of the WebSpective components during run time. The complete listing of configuration parameters can be found in Appendix A. Refer to the section entitled "Configuration" on page 17 for information on viewing and editing these configuration parameters before starting the WebSpective components.

UNIX Installation

Installation on UNIX (specifically Solaris) systems is done with the pkgadd utility. You can either run this directly from the WebSpective CD-ROM, or you can copy the necessary information to your local machine first.

Local Installation

To perform the installation from your local machine, locate the file named “webspective1_0.tar.Z” on the media you received and copy it to a temporary directory. Uncompress and expand the file as follows:

```
uncompress webspective1_0.tar.Z <ENTER>
tar xvf webspective1_0.tar <ENTER>
```

The file expansion creates a new directory titled “/webspect”. This directory contains all of the information that pkgadd will need to install the WebSpective system.

Setting Up Files with *pkgadd*

Log in as root and go to the WebSpective package’s base directory. If you are installing directly from the CD-ROM this is /solaris/webspect. Run pkgadd with the following command:

```
pkgadd -d . -R <load-path> <ENTER>
```

Where *load-path* is the path under which you would like to install WebSpective (usually /usr/local). The utility will warn you that it is making changes to the system as root. Specifically:

- Interceptor_exe and ~~the~~ Agent are setuid to root and setgid to group “nobody”
- Pkgadd will be running scripts as root

For information on disabling setuid and setgid, refer to Appendix D: Security Considerations.

Installation proceeds, and when it has finished, the path to a set of template files appears. These templates are the registry files for each component of the WebSpective system. Copy the template for the components you are installing to a permanent location. Atreve suggests the “etc” directory below the directory in which you installed WebSpective.

Making a K yfil

Before you can run your WebSpective component, you need to provide or generate a keyfile for the system. Packaged with WebSpective is a utility called keygen. Run the keygen utility by entering the following command:

```
keygen [keyfile_name]
```

Where keyfile_name is the name you want to assign to your keyfile (The Atreve standard is to end keyfile names with a ".key" extension). Put the keyfile in the desired location.

Once you have installed the components and prepared a keyfile, you are ready to edit the registry file of the component you wish to run. Refer to the section titled "Configuration" on page 17 for information on editing registry files.

Setting Environment Variables

Before WebSpective will run properly, the following variables must be set in the user's shell. In these listsings, *load-path* refers to the path under which you installed the WebSpective system.

- **LD_LIBRARY_PATH**—must be set to point to the location of the WebSpective shared libraries. In k-shell, this would be done as follows:

  ```
  export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:<load-path>/atreve/
  lib"
  ```

- **NLS_LANG**—If you are using Oracle's JDBC driver (which is recommended), then you will need to set this environment variable. In k-shell, this would be done as follows:

  ```
  export NLS_LANG=american_america.US7ASCII
  ```

- **NLSPATH**—must be set to point to the location of the WebSpective message catalog files. In k-shell, this would be done as follows:

  ```
  export NLSPATH="$NLSPATH:<load-path>/atreve/etc/%N.cat"
  ```

- **PATH**—must be set to point to the location of the WebSpective executables. In k-shell, this would be done as follows:

  ```
  export PATH="$PATH:<load-path>/atreve/bin"
  ```

Opti nal Variables. In addition to the required environment settings, you may need to specify additional variables according to your system's configuration:

- **JAVA_HOME**—This determines where WebSpective will look for the Java Run-time Environment (JRE). The default is "/usr/java". Setting this variable will override the default. In k-shell, this would be done as follows:

```
export JAVA_HOME="$JAVA_HOME:<load-path>"
```

- **JAVA_CMD**—This determines which JAVA VM will be used. The default is "jre". Other useful alternatives are "java" and "java_g". In k-shell, this would be set as follows:

```
export JAVA_CMD="$JAVA_CMD:<vm_name>"
```

SSL Support

WebSpective makes use of a Java web server in providing SSL support on UNIX-based systems. You will need to install this web server on any UNIX machine that is going to host secure servers under WebSpective control.

Once you have expanded the Java web server's tar file, you will need to edit the following variables in the user's shell. In these listsings, *load-path* refers to the path under which you installed the web server:

- **CLASSPATH**—must be set to point to JavaServer's classes. In k-shell, this would be done as follows:

```
export CLASSPATH="$CLASSPATH:<load-path>/JavaServer1.0.1/
lib/classes.jar"
```

- **LD_LIBRARY_PATH**—must be set to point to the location of the JavaServer shared libraries. In k-shell, this would be done as follows:

```
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:<load-path>/
JavaServer1.0.1/lib/solaris/sparc
```

Filter Installation

The Filter is an NSAPI or ISAPI program that watches hits passing through the web server and sends this hit information to the Agent. The Filter itself is a shared library that must be loaded with, and effectively become part of, the web server. The method for applying the filter to the server varies according to each vendor's specifications. Currently, WebSpective supports the Netscape Enterprise and FastTrack servers (versions 2.0 or higher), as well as Microsoft IIS servers.

Installation on a Netscap FastTrack or Ent rprise Server

Netscape servers refer to an obj.conf file for direction on handling transactions. This file, typically located in the server's "<loadpoint>/https-server-name/config" directory, must be edited to include calls to the Filter. There is a distinct order in the obj.conf file that the server follows in processing hits. In following this order, inclusion of the Filter involves four steps:

1. Initialization
2. Name translation
3. Service routing
4. Logging

When you open your obj.conf file, you will see that it is divided into a number of sections:

```
#Netscape Communications Corporation - obj.conf
#You can edit this file, but comments and formatting changes
#might be lost when the admin server makes changes.

#Use only forward slashes in pathnames--backslashes can cause
#problems. See the documentation for more information.
```

"Init" Section

```
Init format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%] \"%Req-
>reqpb.clf-request%\" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
fn="flex-init" access="C:/Netscape/Server/https-oolong/logs/access"
Init fn="load-types" mime-types="mime.types"
Init fn="load-modules" funcs="AtreveInitFilter,AtreveInitialFunction,AtreveAddLog-
Function,AtreveServicePassthrough" shlib="d:/users/developer/atreve/devbin/atreve-
flt.dll"
Init fn="AtreveInitFilter" name="https-oolong"
Init funcs="rdm-init,rdm-type,rdm-service,rdm-log" fn="load-modules" shlib="C:/
Netscape/Server/plugins/autocatalog/lib/rdm.dll"
Init config="C:/Netscape/Server/https-oolong/config/rdm.conf" fn="rdm-init"
```

"default" Object Section

```
<Object name="default">
NameTrans fn="AtreveInitialFunction"
NameTrans from="/ns-icons" fn="pfx2dir" dir="C:/Netscape/Server/ns-icons"
NameTrans from="/mc-icons" fn="pfx2dir" dir="C:/Netscape/Server/ns-icons"
NameTrans from="/cgi-bin" fn="pfx2dir" dir="c:/Netscape/Server/cgi-bin" name="cgi"
NameTrans fn="pfx2dir" from="/NPIdocs" dir="C:/Netscape/Server/docs/NPIdocs"
name="NPIdocs"
NameTrans fn="pfx2dir" from="/catalog" dir="C:/Netscape/Server/https-oolong/catalog"
NameTrans root="C:/Netscape/Server/docs" fn="document-root"
PathCheck fn="nt-uri-clean"
PathCheck fn="find-pathinfo"
PathCheck index-names="index.html,home.html" fn="find-index"
ObjectType fn="type-by-extension"
ObjectType fn="force-type" type="text/plain"
ObjectType fn="rdm-type"
Service ufn="imagemap" fn="AtreveServicePassthrough" method="(GET|HEAD)" type="mag-
nus-internal/imagemap"
Service ufn="index-common" fn="AtreveServicePassthrough" method="(GET|HEAD)"
type="magnus-internal/directory"
Service ufn="send-file" fn="AtreveServicePassthrough" method="(GET|HEAD)"
type="*-magnus-internal/*"
Service ufn="rdm-service" fn="AtreveServicePassthrough" type="application/x-rdm"
AddLog fn="AtreveAddLogFunction"
AddLog fn="flex-log" name="access"
AddLog fn="rdm-log"
</Object>
```

Object Section

```
<Object name="cgi">
ObjectType fn="force-type" type="magnus-internal/cgi"
Service ufn="send-cgi" fn="AtreveServicePassthrough"
</Object>
```

How to edit th "obj.conf" file

1. *Initialization.* In the "Init" section, you have to specify modules (functions) that should be loaded from the filter, and where the filter is. At the end of the Init section, add these lines:

   ```
   Init fn="load-modules" funcs="AtreveInitFilter,AtreveIni-
   tialFunction,AtreveAddLogFunction,AtreveServicePassthrough"
   shlib="<path>"

   Init fn="AtreveInitFilter" name="server" (regfile="<path>")
   ```

   Note that functions listed in the first line are separated by commas (,) and no spaces. The server will not read the line correctly if spaces are included in the list.

   The "shlib" path points to the shared library which supplies those functions. This library is called "libatreveflt.so" on UNIX systems, "libatreveflt.dll" on Windows NT systems, and can be found in the "lib" directory under the directory where WebSpective was installed. Note that when entering this directory path you must use *forward slash* (/) dividers, even on Windows NT systems. Here is an example:

   ```
   C:/Program Files/Atreve/WebSpective/lib/libatreveflt.dll
   ```

   In the second line, the "name" parameter is the name of this server, and must be the same as the name given when the server is created in the Viewer. The "regfile" parameter is needed only on UNIX. It must point to the registry file created by WebSpective when you add a server to the system. The location of this file is governed by a web server's Registry property. see "Registry" on page 35

2. *Name translation.* All of the requests to a given server must first pass through the Filter. Therefore, name translation for the filter must happen before any of the other services. In the "default" Object section, add the following line directly after the <Object name="default"> tag:

   ```
   NameTrans fn="AtreveInitialFunction"
   ```

3. *Service routing.* Client requests must be explicitly routed through the filter before they are handed off to their intended service. A generic service directive might appear as follows:

   ```
   Service fn="imagemap"
   ```

   To direct this service call through the filter, you would change the line to read:

   ```
   Service ufn="imagemap" fn="AtreveServicePassthrough"
   ```

   This syntax tells the server that the client request will go to the "imagemap" function after being sent to "AtreveServicePassthrough".

   Service directives are located throughout the obj.conf file. Any time you add a new service to your web server, you will have to reopen the obj.conf file and redirect any new service calls through the filter. Any service directives that are not sent through the filter will not be reported to the WebSpective system and can potentially skew WebSpective's load balancing performance.

4. *Logging.* To produce a log of a client transaction, add-ons to the server must provide a directive in the obj.conf file that points to a logging function. Towards the bottom of the "default" Object section, there are a number of entries that begin with the "AddLog" directive. The Filter's log, similar to the Filter itself, must be called before other services. Insert the following line so that it is the first Addlog entry in the "default" Object section:

```
AddLog fn="AtreveAddLogFunction"
```

Installation on a Microsoft IIS server

To use the Filter with the Microsoft IIS server, you will need to edit the host machine's registry file. Once you have specified the path to the Filter library in the registry, you will be able to start the server with the Filter intact.

How to install the Filter on the Microsoft IIS server

1. Note the location of the IIS Filter library, called "atreveiisflt.dll". The default location is:

```
C:\Program Files\Atreve\WebSpective\bin\atreveiisflt.dll
```

2. Start the Registry Editor by typing "regedit" at a command prompt.
3. Navigate the registry tree to the key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\
W3SVC\Parameters
```

4. Find the "Filter DLLs" parameter in the list on the right and double-click on it to open the Edit String dialog box.
5. Place your cursor at the beginning of the string and add the path which you noted in step 1. Separate the newly entered path from pre-existing ones with a comma (,). Do *not* delete the other entries in this string. When you are finished, press the OK button at the bottom of the dialog.
6. Exit the Registry Editor. The Filter has been installed.

Testing the Filter installation

Once you have made changes to the obj.conf file or the registry, you can perform two tests to see that the Filter has been correctly installed.

Running *checkns* (Netscape only). The first testing method involves running a script that will read over the edited obj.conf file. Find the script named "checkns" and run it at a command prompt with the path and name of the file you want to test as the argument:

```
checkns /usr/temp/obj.conf
```

The script will read your file and notify you of any lines that still require editing.

Run-Time Test. The second testing method requires you to start the server as part of the WebSpective system, via the Viewer. If the server starts without errors while using your edited obj.conf file, then it is safe to assume that the Filter is working.

If the server fails to start in either case, check the syslog or the Event Viewer for error messages which will help you to debug your installation.

Configuration

Once the WebSpective components have been installed, they should be able to start without any additional configuration. If you need to reconfigure elements which cannot be modified through the Viewer, such as the Interceptor's Manager Port parameter, you will need to run the configuration utility at each component's local machine.

Note that if you are using WebSpective on a UNIX system that you can also manually edit each component's registry file using a tool such as *vi* or *emacs*.

How to run the configuration utility

1. If the component you want to work with is running, shut it down. Refer to page 20 for information on shutting down components locally, and page 42 for information on shutting down components through the Viewer. From a command prompt, type:

   NT: `configure <ENTER>`

   UNIX: `configure` *`reg_file`* `<ENTER>`

   Where *reg_file* is a UNIX-only argument specifying the path to and name of the registry file on the local machine. The configuration utility will appear.

2. If you are working with a machine upon which more than one component has been installed, choose a component to work with by selecting it from the list at the top of the dialog box. All of the parameters for the selected component will appear in the dialog box. If values have been set for any of the parameters, they will appear next to the parameter names.

3. Edit parameter values by clicking in the entry boxes next to the parameter names and typing in the necessary information. A full description and possible values for each parameter can be found in Appendix A.

4. When you have finished editing the configuration parameters, click the Save button at the bottom of the dialog box to write your changes to the registry or the registry file. You can then select another component to edit (if other components reside on the same machine) or press the exit button at the bottom of the dialog box to close the configuration program.

Chapter 3

Operating WebSpective

Chapter Contents:

- Start-Up and Local Control
- The Viewer
- The Management Tab
- The Health Tab
- The Event Tab

Start-Up and Local Control

Once all of the components of the WebSpective system have been configured, you can begin to start them from their individual machines. Webspective has been designed so that its components can be started in almost any order. The only exception to this rule is the Viewer. Because the Viewer gets all of its information from the Manager, the Manager must be running before the Viewer can be started.

In the interest of bringing up the WebSpective system in the shortest amount of time, Atreve recommends that you start the Manager first. If the Manager is running, other components will be able to register with it as soon as they are started.

How to start a WebSpective component

On Windows NT systems, the Agent, Interceptor, and Manager are services which start when the system is started and can be controlled locally through the Services Control Panel. The Viewer can be started by selecting it from the Start menu.

On UNIX systems, type the following at a command prompt on the component's local machine:

```
component_name (reg_file
```

Where *component_name* is the name of the component you wish to start (in all lowercase letters) and *reg_file* (for UNIX systems only) is the name of the file that contains the component's configuration parameters.

Assuming you have started the Manager and the Viewer first, you will see other components appear in the viewer as they start. While the Viewer offers you full control over your web site, you can still stop WebSpective components locally if necessary.

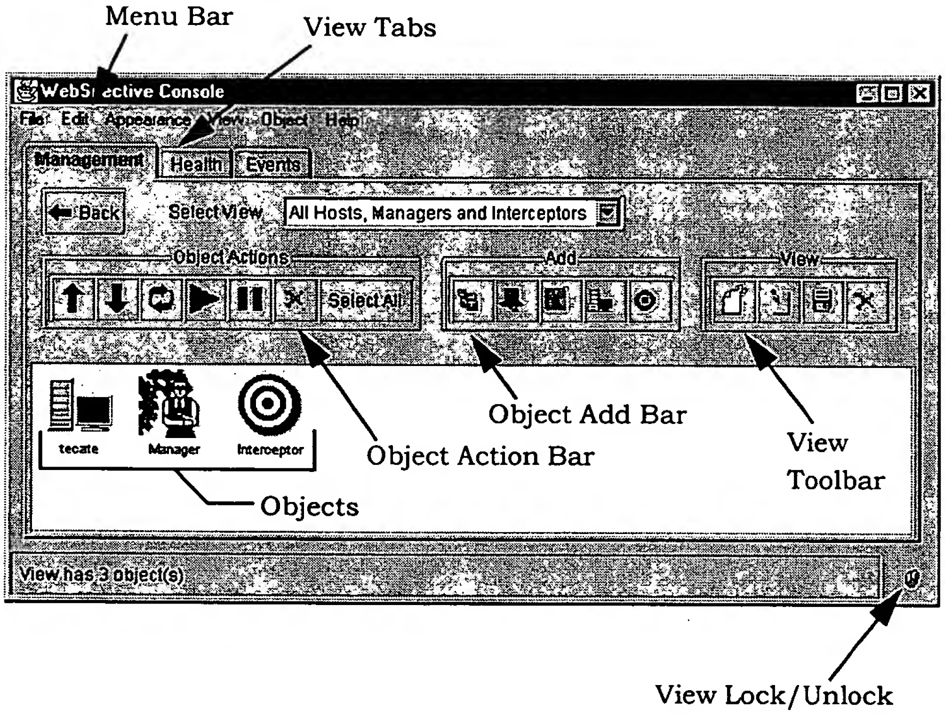
Shutdown

WebSpective components can be shut down in a few different ways. First, you can stop them through the Viewer. Next, on UNIX systems, components can be stopped with a plain "kill" command either to their watcher process or to their main process. Finally, on Windows NT systems, the Manager, Agent, and Interceptor can be stopped from the Services Control Panel.

We recommend attempting to stop WebSpective components via the Viewer before shutting them down locally.

The Viewer

The Viewer is the user's primary interface to the WebSpective system during normal operation. Its interface consists of a menu bar, three tabbed panels, a toolbar, and a display area as shown below.



View has 3 object(s)

Navigating with the Mouse

To examine the contents of any object, double-click on it. The view will change to display the children of the chosen object. For example, if you were to double-click (or *drill down*) on an application called "Atreve Software", the current view would be replaced with a view of the endpoints that support it. In this fashion, you can move through the hardware- and software-based organizations of your web site.

Right-clicking on objects in the site field causes a pop-up menu to appear. This menu contains all of the functions that can be applied to the selected object. See Chapter 4 for more information on object functions.

The Management Tab

The Management Tab provides the user with a graphic representation of the entire WebSpective-controlled web site. Icons in this view represent the site at every level, from web applications down through the specific endpoints that carry them.

Exploring the Web Site

By default, all of the hosts and applications in your web site are visible in the site field. You can modify this view by choosing a selection from the combo box in the Views toolbar.

The three initial views are:

- **All** (the default view, which shows all hosts and applications)
- **All Applications**
- **All Hosts**

Creating Custom Views

In addition to the initial views, you can create your own views to show any combination of elements on your web site. You can also move icons around the screen to help organize them as you see fit.

How to create a new view

1. Press the **New** button in the Views toolbar. The Management View Configuration window appears.
2. Type a name for the new view in the **Name of View** box in the upper left corner of the Management View Configuration windows.
3. In the **Select Objects** field, pull down the menu in the **Type** combo box to choose the kind of object you would like to add to your view. For the object type you select, a list of items will appear in the field below the combo box.
4. Single-click the specific items you would like to add to your view. Click on the right-pointing arrow to in the middle of the window to add the selected items to the **Current View Members** list. To remove selected objects from the Current View Members list, single-click on them and press the left-pointing arrow to the left in the middle of the window.
5. When you have finished adding items to your customized view, click on **OK** to save and activate it in the Viewer.

Note that there is no limit the number or type of items you can add to your view.

How t dit a cust m view

To edit a custom view, select it as the active view and press the **Edit** button in the Views toolbar. The Management View Configuration window appears. Follow the instructions in steps 2 through 5 of "How to create a new view" to make changes to the custom view.

How to save a view under a new name

Once you have created a view, you may want to use it as a template for other views. Select the view you wish to copy as the active view and press the **Save As** button in the Views toolbar.

How to delete a view

1. Select the view you wish to delete as the active view and press the **Delete** button in the Views toolbar.
2. The Viewer will ask you to confirm the request. Press the **OK** button to continue or **Cancel** to stop the request.

The Creation Bar

In the Appearance menu at the top of the Viewer, you can toggle a hide/show option for the Creation Bar. When activated, the Creation bar appears just above the site field.

Active buttons on the Creation bar correspond to the objects that can be created in a given view. For example, when you select the All Applications view, the Application button becomes active in the Creation bar. To learn more about the creation of objects, see Chapter 4: Site Management.

The Health Tab

Like the Management Tab, the Health Tab allows you to configure various views of the objects in your website. Unlike the Management Tab, however, the Health Tab uses different graphs to represent the vital signs of the objects you select.

Preconfiguration for Windows NT-based Agents

Before you can use the Health View on a system with NT-based Agents, you will need to enable disk and network performance monitoring on the Agents' host machines.

How to enabl disk and network performanc m nit ring

1. At a command prompt, type:

```
diskperf /y <ENTER>
```

This enables disk performance monitoring.

2. Open the Control Panel and double click on the Network item. The Network interface appears.
3. In the Network interface, switch to the Services tab. Click on the Add button in the Services tab and choose "SNMP Service" from your list of options.
4. Configure the new SNMP service to report network interface statistics and close the network interface.
5. Reboot the machine. When it restarts, you will be able to use the Health view with that machine.

View, Panel, and Graph Creation

Each *view* in the Health Tab is comprised of *panels* that contain one or more *graphs*. The function of panels is to define preset graphs and graph groups that you can quickly add to your views. Once a panel has been created, you can call it into use with any new view you make.

How to create or edit a view

1. To create a new view, press the New button from the Health View toolbar. To edit an existing view, choose the layout you wish to edit from the Health View toolbar and press the Edit button. The Health View Layout Editor appears.
2. Determine or modify the organization of your view by using the buttons in the lower left of the Layout Editor:

   Split Horizontal—splits the selected area horizontally into two equal areas
   Split Vertical—splits the selected area vertically into two equal areas
   Remove Split—recombines the areas divided by the selected divider

   Continue to divide areas by selecting them and choosing one of the split options until you have created the desired layout. Note that in addition to splitting areas, you can resize them by clicking and dragging the section dividers.
3. Add or change panels to each area of the view. To do this, select an area and choose a panel from the combo box at the bottom of the screen. If no panels have been defined, or if you want to create a new one, see the section below, titled "How to create or edit a panel".
4. If you are creating a new view, enter a name for it into the box at the top of the Layout Editor. Press the OK button. The view is automatically saved.

How to create or edit a panel

1. Create a new panel by pressing the New button at the bottom of the Layout Editor. Edit an existing panel by selecting it from the combo box at the bottom of the Layout Editor and pressing the Edit button. The Panel Setup window appears.
2. Determine the number of rows and columns the panel will contain. To do this, adjust the numbers in the Row and Column boxes at the top right of the Panel Setup window.

3. For each cell in the panel, add a graph by pressing the Add button at the bottom of the Panel Setup window, or edit an existing graph by pressing the edit button. See the section below, titled "How to create or edit a graph", for more information.
4. Once you have finished adding graphs to the panel, press the OK button at the bottom of the window. The panel is automatically saved.

How to create or edit a graph

1. Press the Add button at the bottom of the Panel Setup window or select a graph from the list in the Panel Setup window and press Edit. A dialog box appears asking you what type of graph you would like to create.

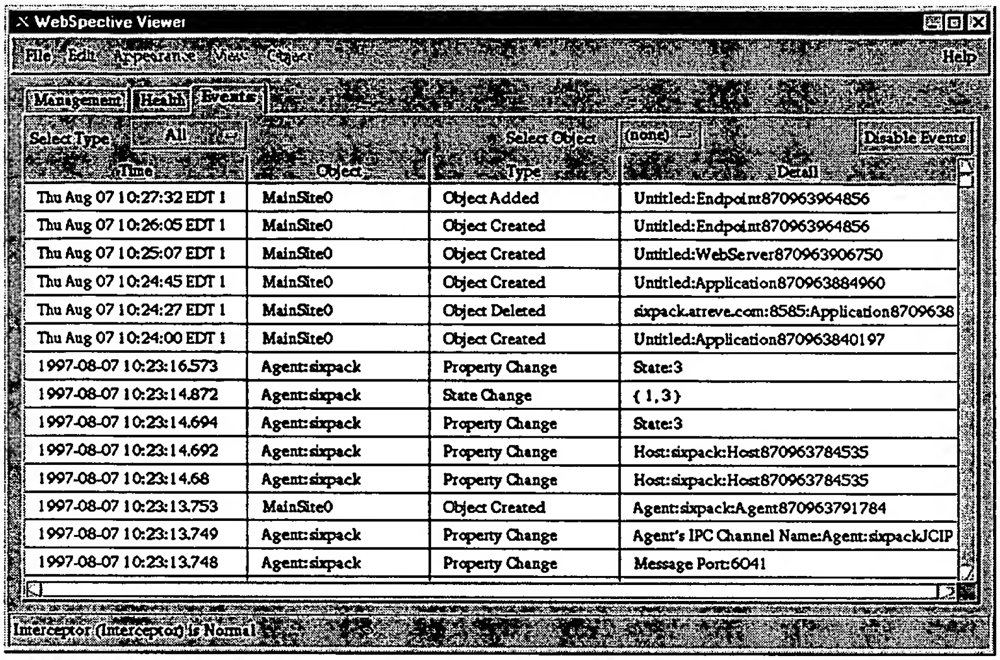
   Currently, the only type of graph you can choose is a Rolling graph, which plots one object versus a statistic over time.
2. Choose an object to monitor by selecting its type from the pulldown menu and clicking on the specific object from the list on the left side of the Create Rolling Graph window. Components of the selected object (if applicable) appear in the list in the center of the window. Possible statistics for the selected object appear in the list on the right side of the window.
3. Choose a component (if applicable) and a statistic from their respective lists and press OK.
4. If you would like to further constrain the results of the graph, check the Advanced Parameters checkbox. A text box appears, into which you can enter SQL code that will be appended to the generated query. Refer to Appendix B for information on the organization and syntax used by the logging databases.

   An example would be to track hits to a specific URL:

   ```
   where FSpath = '/products/index.html'
   ```

The Event Tab

The third tab in the Viewer interface, the Event tab, provides the user with a continuously updating list of events within the WebSpective system:



Time	Object	Type	Detail
Thu Aug 07 10:27:32 EDT 1	MainSite0	Object Added	Untitled:Endpoint870963964856
Thu Aug 07 10:26:05 EDT 1	MainSite0	Object Created	Untitled:Endpoint870963964856
Thu Aug 07 10:25:07 EDT 1	MainSite0	Object Created	Untitled:WebServer870963906750
Thu Aug 07 10:24:45 EDT 1	MainSite0	Object Created	Untitled:Application870963884960
Thu Aug 07 10:24:27 EDT 1	MainSite0	Object Deleted	sixpack.atreve.com:8585:Application8709638
Thu Aug 07 10:24:00 EDT 1	MainSite0	Object Created	Untitled:Application870963840197
1997-08-07 10:23:16.573	Agent:sixpack	Property Change	State:3
1997-08-07 10:23:14.872	Agent:sixpack	State Change	{ 1,3 }
1997-08-07 10:23:14.694	Agent:sixpack	Property Change	State:3
1997-08-07 10:23:14.692	Agent:sixpack	Property Change	Host:sixpack:Host870963784535
1997-08-07 10:23:14.68	Agent:sixpack	Property Change	Host:sixpack:Host870963784535
1997-08-07 10:23:13.753	MainSite0	Object Created	Agent:sixpack:Agent870963791784
1997-08-07 10:23:13.749	Agent:sixpack	Property Change	Agent's IPC Channel Name:Agent:sixpackJCIP
1997-08-07 10:23:13.748	Agent:sixpack	Property Change	Message Port:6041

The following is a list of the kind of events that will appear in the Event Tab.

- Data Samples
- Security Violations
- Threshold Events:
  - Variable Over Threshold
  - Variable Under Threshold
  - Variable Over High Threshold
  - Variable Under Low Threshold
- State Changes
- Ping Events:
  - Ping Timedout
  - Ping Failed
- Property Changes
- Host Events:
  - Host Up
  - Host Down
  - Host Problem
  - Host Activate
  - Host Deactivate
- Safe Stops
- Application Events:
  - App Up
  - App Down
  - App Problem
  - App Activate
  - App Deactivate
- Object Events:
  - Add Object
  - Add Member
  - Rem Object
  - Rem Member
  - Create Object
  - Delete Object
- Load-Balancing Events:
  - Epsilon Add To App
  - Epsilon Remove From App
  - Epsilon Activate
  - Epsilon Deactivate
- Error Events

You can change the sorting criteria of the table by clicking the title of any column. When you do so, the table will sort itself according to the data in the column you selected.

Chapter 4

Site Management

The Viewer provides you with full control over the WebSpective-managed web site. You can work with objects at any level, performing a wide array of tasks. This chapter is arranged by object to help you quickly find help with site operations.

Chapter Topics

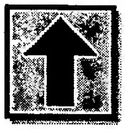
- Object States
- Object Properties
- Object Management

Object States

In the following sections you will find functions and procedures for each of the objects that may be seen in the Viewer's Management Tab. While you perform functions on a given object, it will likely go through one or more states.

There are ten states in which an object can exist. Each change of state is reflected in the Viewer. When an object's state changes, its icon is overlaid with an indicator as illustrated below.

A state in one object will propagate to all contained objects. For instance, if a web server is Deactivated, all of its endpoints will similarly be deactivated.



Starting. During creation and after periods of Deactivation or being Off, an object must be started and registered with the Manager and Agents' Object Databases (ODBs). While this is happening, the object is in the Starting state and is not yet operational. Once the initialization process is complete, the object moves into the Normal state.

Stopping. Entities such as web servers cannot stop "on a dime"—they must go through a period in which clients can be allowed to complete transactions and in which instructions are sent to relevant parts of the WebSpective system. When an object has finished stopping, it is Off.

Deactivated. As opposed to an object that is Off, a Deactivated object is still running, and most likely actively directing hits back to the Interceptor. The Interceptor itself can also be deactivated, in which case clients who hit the Interceptor will be redirected to a URL specified in the Application's properties (see "Deactivated URL" on page 32).

Problem. The Problem state indicates that something has gone wrong with the affected object, but that the object is still partially functioning. Remember that states propagate—if an endpoint enters a Problem state, the web server that supports it, as well as the application that owns it, will enter the problem state as well. Specifically, an object enters the Problem state when at least one of its contained objects is either Problem or Failed and at least one of its contained objects is still Normal.

Restarting. Unlike the Starting state, Restarting only occurs when an object goes down unexpectedly and is automatically restarted by WebSpective. As in the case of web servers, the WebSpective system will make a number of attempts to restart an object before giving up and reporting that the object has Failed.

Failed. After a certain number of attempted restarts, or in situations where a an object cannot function even partially, that object enters the Failed state. Objects that have failed will require your direct attention and cannot be serviced from inside the Viewer. In addition to the direct failure, an object will automatically go to Failed when all of its contained objects go to Failed.

Unknown. Unknown objects are objects that are recognized by the WebSpective system, but can not connect to it. WebSpective is unable to determine the state of an Unknown object and cannot open any channels of communication. This is likely to happen with an Interceptor that has been configured outside of the currently operating WebSpective system.

Redirecting due to Load. When a server has reached the top of its request-handling capacity, it enters the Redirecting state. In this state, the web server begins sending requests back to the Interceptor and is also removed from the Interceptor's availability listing. When the server's hit count drops below a certain point, it drops back to the Normal state and is re-added to the Interceptor listing.

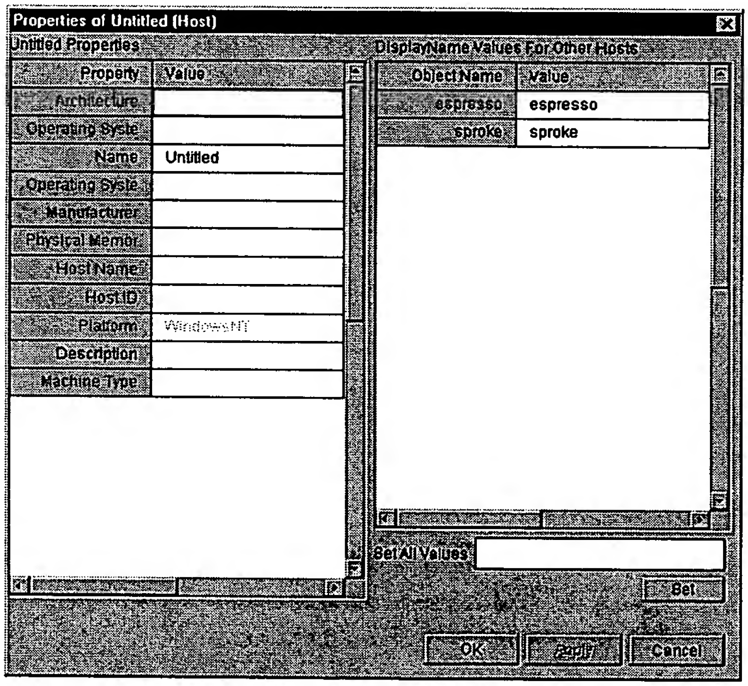
Maxed Out. After a user-defined number of tries, the Agent gives up on restarting servers that have gone down. Once this happens, the server is considered Maxed Out. A server in this state may be restarted by the user via the Viewer after the problem has been found, at which point the Maxed_Out counter is reset by the system.

Normal. After an object has been successfully started, it will remain in the Normal state until it is manipulated by the user or a problem arises in the system. The icons for objects in this state have no icon overlay.

Off. Objects in the Off state are simply Off. The process the object represents is not running. Objects that are Off appear grayed out in the interface.

Object Properties

Each object has distinct properties that are set when the object is first installed. Things like IP addresses, port numbers, and host names are established before components are first brought on-line and cannot be changed through the Viewer. However, other properties can be configured during runtime. When you right-click on an object, you will see a pop-up menu. The last entry in the pop-up menu is *Properties*. Selecting this item opens the Properties Editor:



Properties of Untitled (Host)

Untitled Properties

Property	Value
Architecture	
Operating Syste	
Name	Untitled
Operating Syste	
Manufacturer	
Physical Memor	
Host Name	
Host ID	
Platform	WindowsNT
Description	
Machine Type	

DisplayName Values For Other Hosts

Object Name	Value
espresso	espresso
sproke	sproke

Set All Values:

Items that appear in blue (such as the *Name* property in the above dialog) are required items. Items that appear in black (such as the *Description* property) are optional items. Items that appear in gray (such as the *Platform* property) are unchangeable.

Note that all of the objects of the same type are listed in the list on one side of the dialog box. This allows you to make changes on more than object at the same time. For instance, if you change the location of the KeyFile on a host, you will want to update all of the web servers on the host with that information. You can do this either by editing the values in the grid, or by using the "Set All Values" field.

How t change a pr perty using “S t All Valu s”

1. Select an object and right-click to call the actions menu.
2. Choose **Properti s** from the menu. The Properties Editor appears.
3. In the field on the left side of the dialog box, choose the property that you would like to change across all like objects and enter the appropriate value. The *Set All Values* entry box in the lower right portion of the dialog box automatically reflects the new value.
4. When you are satisfied with the new global value, click the **Set** button below the Set All Values entry box to change the property in all of the like objects, and press the OK button to save the change.

Note: At this time, the ability to select only certain objects to update out of the entire set of like objects is currently unavailable.

Property Definitions

Each object has configurable properties that are specific to the object type in addition to a few properties that all objects have. The following is an alphabetical list of properties and their definitions.[1]

AgentLocalPort

Value: Integer

Used by: web servers

The port number at which the Agent listens for messages from the Manager.

Audit Commands

Value: True or False

Default: True

Used by: Interceptor

When set to “True”, the Interceptor logs *all* commands to the logging viewer specified in Mgmt Log.

AppIPAddr

Value: Character string or a number in dotted-decimal notation

Used by: Applications

The IP address at which the Interceptor listens for hits to the given Application.

AppPort

Value: Integer

Used by: Applications

The port number that the Interceptor listens to for hits to the given Annlication.

[1] If you intend to use the Interceptor Control Driver after setting Interceptor properties in the Viewer, make sure that you follow the restrictions the Control Driver requires in setting properties. Specifically: property values cannot contain spaces, property values cannot exceed 64 characters in length, and only ASCII standard characters can be used in property values.

CloseDelay

Value: Integer

Default: 0 on Windows NT systems, 100 on Sun Solaris systems

Used by: Interceptor

The time in milliseconds to wait before closing a connection that a client has opened.

Command TTL

Value: Integer

Default: 60

Used by: Interceptor, Agents, web servers

The time, in seconds, that a command sent from one object to another should be considered valid. This allows for discrepancies between the systems clocks at both ends of the transmission. However, if the Command TTL is set too high, you increase the possibility of "replay" attacks on your website.

Comment

Value: Character string

Used by: All objects

A description of the object.

Deactivated URL

Value: Character string

Used by: Applications

A URL that points to a web page to be served in place of a Deactivated application.

DecayWindow

Value: Integer

Default: 600

Used by: Interceptor

The amount of time in seconds that the interceptor will consider any one set of load data accurate. If the Manager does not send an update before the DecayWindow expires, the Interceptor will return to the default settings that it loaded on start-up.

Hardware

Value: Character string

Used by: hosts

A field used to describe the host's processor architecture.

Host

Value: Character string

Used by: Manager, Interceptor, Agents, web servers, Endpoints

The name of the host upon which the object resides.

Host ID

Value: Character string

Used by: hosts

This value uniquely identifies the host machine (without regard for the machine's assigned host name). This string is generated outside of WebSpective and cannot be changed.

Host Name

Value: Character string

Used by: hosts

The actual name of the host, versus the Name property that can be more descriptive if necessary.

IPCchannel

Value: Character string

Used by: Agents

The name of the shared memory channel that the Agent uses to communicate with web servers.

IPname

Value: Character string

Used by: Endpoints

The name of the host upon which the Endpoint is running. Note that this value *cannot* be entered in dotted-decimal notation. In order to be verified under the Interceptor's "AccDomains" configuration parameter, the name of the endpoint's host must be provided.

IsService

Value: True or False

Used by: web servers

Reflects the web server's operation as a service (on Windows NT).

KeyFile

Value: Character string

Used by: Manager, Interceptor, Agents, web servers

The path to the WebSpective-generated keyfile. Without the keyfile, components of the WebSpective system cannot communicate with the Manager and vice versa. Web servers will default to their Agent's keyfile if one is not specified, however there is no default for other components. Refer to Appendix D: Security Considerations for more information on the Keyfile.

LingerTimeout

Value: Integer

Default: 20

Used by: Interceptor

The number of seconds that the Interceptor waits before closing a connection with an inactive browser. This protects the Interceptor from "denial of service" attacks.

Logging Fi lds

Value: Integer

Used by: Endpoints

Allows you to determine what kind of information is polled from hits to the Endpoint and recorded to the LDB. For information on how to set the logging fields, refer to the section titled "Logging Fields" on page 38.

Mgmt IP/name

Value: Character string or a number in dotted-decimal notation

Used by: Interceptor, Agents, web servers

The name or IP address at which the manager listens for messages. Note that this can be on a separate network which is isolated from website traffic by using a second network card to carry it.

Mgmt Log

Value: Character string

Used by: Manager interceptor, Agents, web servers

Points to where messages are logged by the Platform—syslog on Unix and the Event Viewer on Windows NT.

Mgmt Port

Value: Integer

Used by: Interceptor, Agents

The port number at which the Manager listens for messages.

MgtInterval

Value: Integer

Used by: Interceptor, Agents, web servers

The frequency, in seconds, with which the object is pinged. All of the specified ping URLs on a given object are tried during each interval. Currently, the Interceptor and Agents do not use this value, however, it will be implemented in future releases.

NOTE—

Ping processing against an SSL server can use up to 100% of the available CPU time. Consequently, you should not set the MgtInterval property of a secure server to less than 60 seconds. This ensures that CPU time will be available for the delivery of server content.

Name

Value: Character string

Default: "Untitled"

Used by: All Objects

The name of the object. Atreve Software, Inc. recommends the following conventions for naming various objects:

Agents—agent:<host name> (agent:espresso, for example)

Endpoints—<host name>:<port> (espresso:80, for example)

Web Servers—follow the standard used by the web server's manufacturer. For Netscape servers, the name must match that found in the server's obj.conf file, as discussed on page 15.

Operating System Revision

Value: Character string

Used by: hosts

Describes the current version of the host's operating system

Physical Memory

Value: Integer

Used by: hosts

The amount, in megabytes, of physical RAM in the host machine. This value is automatically detected an cannot be changed.

PingTimeout

Value: Integer

Default: 30

Used by: Endpoints

The time in seconds that the Agent will wait before giving up on a ping to the given Endpoint.

Platform

Value: Character string

Used by: Agents, Applications, hosts, web servers

Describes the operating system upon which the object is running. This value is automatically detected an cannot be changed.

PortTakeover

Value: True or False

Used by: Interceptor

This property determines how the Interceptor will respond on startup when the port it has been assigned is not available. When set to "True", the Interceptor will wait for the port to become available. When set to "False", the Interceptor will log a failure and will not start.

RecvTimeout

Value: Integer

Default: 5

Used by: Manager, Interceptor, Agents, web servers

The number of seconds that an object should wait before closing an inactive connection.

RedirectURL

Value: Character string

Used by: Endpoints

The URL of a web page that is served when the endpoint has been Deactivated.

Registry

Value: Character string

Used by: Manager, Agents, web servers

On Windows NT systems, this property specifies the location in the Registry where configuration information can be found for the given object. On UNIX systems, this property points to a file that contains the configuration information.

ReviveInterval

Value: Integer

Default: 120

Used by: Manager, Interceptor, Agents, web servers

The number of seconds that the Agent should wait between attempts to start a failed process. The Manager and Interceptor carry this property as well but currently do not use it.

ReviveRetries

Value: Integer

Default: 5

Used by: Manager, Interceptor, Agents, web servers

The number of retry attempts the Agent should make before giving up on a process. Each attempt is spaced ReviveInterval seconds apart.

SecureServer

Value: True or False

Default: False

Used by: web servers

Indicates whether or not the selected web server is a secure (SSL) server.

ServiceName

Value: Character string

Used by: web servers (NT only)

The web server's service name, as determined by the server upon installation.

SetupTime

Value: Integer

Default: 15

Used by: Manager, Interceptor, Agents, web servers

The amount of time (in seconds) that the Agent should wait before attempting to ping a web server that has just been started. If the value is too low, the Agent will attempt to ping the server before it is ready to respond.

ShutdownURL

Value: Character string

Used by: Interceptor

The URL of a web page to be served when the whole web site has been Deactivated.

SSLPassword

Value: Character string

Used by: web servers

If the object is a secure web server, this field contains the password WebSpective must use to communicate with it.

Security Note:
As with all other data, the SSL Password is passed between WebSpective

components in a non-encrypted format. However, the SSL Password is *only* passed on server startup, and not during regular Manager-server communication.

StartDir

Value: Character string

Used by: Agents, web servers (on UNIX systems)

The directory in which the object must be run.

Startup CmdLine

Value: Character string

Used by: web servers

The command-line entry (with arguments, if any) used to start the web server. Note that for UNIX-based servers, WebSpective adds a "-i" argument to the startup command by default. In doing so, the Agent is able to detect a server failure the moment it happens, as opposed to when the regular ping check is done.

State

Value: Character string

Used by: All Objects

This property reflects the current state of the object, as defined in the section titled "Object States" on page 28. While this property is visible in the Properties Editor, it can only be set using the Action toolbar.

Strength

Value: Integer

Default: 5

Used by: Endpoints

The relative strength of the Endpoint (on a scale of 1 to 10) as it compares to other Endpoints in the same Application. This is used to reflect both the power of the machine and the capacity which is not used by other programs. Thus if one machine is much more powerful than another, but has a number of processes running on it in addition to the endpoint it supports, then it may have a lower Strength than the less powerful computer. This value is only used by the Interceptor when it stops getting updates from the Manager.

Threads

Value: Integer

Used by: Applications

This is the number of requests for an application that the Interceptor can simultaneously process. If the thread count is high, more requests can be handled, but more resources are being kept from other applications. If the thread count is low, the application does not use up resources, but may also keep browsers waiting.

The minimum number of threads is 1, and the maximum number of threads is determined by the OS upon which the Interceptor is running.

TimeRevived

Value: Character string

APPENDIXES

Appendix A: Registry Parameters

This Appendix contains a listing of all of the registry parameters used by the WebSpective System. The parameters are organized alphabetically by component.

Agent Registry

Backup Interval

Optional: Integer

Default: 300

The Agent will backup its object database (ODB) at this interval (in seconds). If the ODB has not changed over the interval, a backup is not made.

ChannelName

Required: Name of the shared-memory IPC channel

The name of the shared-memory IPC channel which the Filter will use to send hit information to the Agent. Typically, all interfaces of all servers on a given host will use the same IPC.

Solaris systems only allow six processes to access a shared-memory channel at a time. Regardless, if multiple Agents are running on a host (which is an unusual circumstance), each Agent must have its own channel.

Database

Required: Directory path to the Object Database (ODB)

This parameter specifies the location of the Agent's ODB file. This database is a subset of the database used by the Manager. It contains information pertinent to the servers and applications which have been established on a given Agent's host.

KeyFile

Required: Absolute path to keyfile

This is a required parameter. This parameter's value is the name of a file to use as a secret key. The key itself is generated during the installation and must be identical for each of the various WebSpective components. The keyfile is not used in processing HTTP requests.

LocalIP

Required: Agent's IP address or DNS listed name

This parameter specifies the IP interface at which the Agent listens for connections from the Manager.

LocalP rt

Optional: Port number

Default: 4040

The IP port on which the Agent listens for connections from the Manager.

Manag rIP

Required: Manager's IP address or DNS listed name

This parameter specifies the IP interface at which the Manager listens for connections from other WebSpective components.

ManagerPort

Optional: Port number

Default: 4040

The IP port on which the Manager listens for connections from other WebSpective components.

Name

Required: Text string

The "name" parameter is used to identify individual Agents. On UNIX systems, this parameter is kept inside the Agent's configuration file. On Windows NT systems it is included in the registry.

Application Registry

An application is defined simply as a collection of Web content, which may be offered by multiple equivalent servers (or virtual servers). Their parameters are stored in a group named "Applications," and within that in sub-sections by the name of the application. The parameters for an application stored at the Interceptor are:

AppLocalIP

Required: IP address

The IP interface on which the Interceptor will receive HTTP requests for this application's HTML content.

AppLocalPort

Required: Port number

The port on which the Interceptor will receive HTTP requests for this application.

Description

Optional: Text string

An optional string, for user reference, describing the nature of the application.

Enabled

Required: True (1) or False (0)

Indicates whether this application is currently being published. If it is not, then the Interceptor will return the contents of the "sorry page" for the application, as specified below.

Endp ints

A sub-section listing servers offering this application's content. Inside the "Endpoints" sub-section are zero or more sections named for servers carrying the application. Each server section contains the following keys:

Enabled

True (1) or False (0) value that identifies whether the server is currently operating.

Epsilon

A measure of the "cost per hit" to this server, used in load-balancing equations.

Load

The last measure of the percentage of server in use, as reported by the Manager.

ServerIP

The IP interface on which this server listens.

ServerPort

The port on which this server listens.

Strength

A measure of the processing power of this server relative to the others in the application.

Time

The time that "Load" was last updated.

SorryPage

Optional: URL

This is a URL to return if a request arrives for the application when either it is disabled, or no servers are available. If the URL begins as "file://", then the content is returned by the Interceptor. This is used if the site is shutdown.

If the URL is not a local file, then the client is given an ordinary redirection to that location, and there must be a server present to handle it. This is used to replace individual servers. If no SorryPage is set at all, then a brief generic "503" message is returned, indicating that a server is unavailable.

Threads

Optional: Integer

This parameter allows the user to control how many threads a given application will use. If a request arrives while all threads are busy, it will sit in the TCP queue until a thread is free to accept it, or until it times out.

Note that different operating systems have different limits for the numbers of threads any one process can have. Be sure in setting a number that you do not exceed the number that your operating systems supports.

Filter Registry

Ag ntIP

Required: IP address

The IP address on which the local Agent's command channel is listening.

AgentPort

Required: Port number

The port on which the local Agent's command channel is listening.

ChannelName

Required: Name of the shared-memory IPC channel

The name of the shared-memory IPC channel which the Filter will use to send hit information to the Agent. Typically, all interfaces of all servers on a given host will use the same IPC.

Solaris systems only allow six processes to access a shared-memory channel at a time. Regardless, if multiple Agents are running on a host (which is an unusual circumstance), each Agent must have its own channel.

KeyFile

Required: Absolute path to keyfile

This is a required parameter. This parameter's value is the name of a file to use as a secret key. The key itself is generated during the installation and must be identical for each of the various WebSpective components. The keyfile is not used in processing HTTP requests.

Mask

Required: Hexadecimal integer

This parameter identifies a required bit-field mask of data elements to log. The mask is computed by adding the number assigned to each element. It is an integer, expressed in hexadecimal. For example, the value "0x7ffffff" includes every element. normally, this value is configured through the Viewer.

RulesFile

Required: URL listing

A required list of URLs from which a client cannot be redirected. Pages included in this list represent a transaction with server-side state, which cannot be transferred to another location.

Rule entries must be absolute paths, and are expressed in terms of the client's URI basename, not the server filesystem. For example:

```
/cgi-bin/buystuff.cgi
```

and not:

```
buystuff.cgi

/usr/netscape/docs/cgi-bin/buystuff.cgi

/cgi-bin/buystuff.cgi?param=value
```

Partial URLs and wildcard specifications are syntactically permitted, but will never match anything. It is possible that the requested URL will be a directory, so if the default page is transactional then the rules file should include both "/directory/" and "/directory/index.html," for example.

Int rceptor Registry

AccDomains

Optional: Key (a text string) and DNS name

When set, the Interceptor will not accept the introduction of servers outside of the given domains. The AccDomains parameter is a section, inside of which are keys and DNS names. The key "Atreve Software, Inc.," for instance, might have the value "atreve.com," in which case servers hosted at ws3.atreve.com would be permitted service. If a value is not set for this parameter, you will be able to add any servers under any domain name to the system.

AcceptSSL

Optional: True (1) or False (0)

Default: 0

This parameter specifies whether or not the Interceptor will accept SSL requests. By default, it does not. Note that other SSL related parameters will not work unless this is made true.

AuditCommands

Optional: True (1) or False (0)

Default: 0

This parameter controls the level of system logging. If this parameter is made true, then all commands received by the interceptor are logged with the client IP address, time and date stamp.

CertificateFile

Optional: Absolute path to SSL certificate

Parameter containing the path to the file containing the SSL certificate for your site. Note that this must be the same certificate that your servers use, in order for the Interceptor to pass requests to them seamlessly. Also note that the AcceptSSL parameter must be set to true for the Interceptor to handle SSL requests.

CloseDelay

Optional: Integer

Default: 0 on Windows NT systems, 1 on UNIX systems

This optional parameter specifies a time to wait, in seconds, before closing a connection at the end of a session. On some TCP/IP implementations, allowing the client to close first frees resources faster.

CloseTimeout

Optional: Integer

Default: 5

This optional parameter specifies the number of seconds which the interceptor will wait before closing a connection to a client, after writing data to that client. Note the difference between this parameter and RecvTimeout, which is an equivalent waiting period to read data from the connection

DecayWindow

Value: Integer

Default: 600

Used by: Interceptor

The amount of time in seconds that the interceptor will consider any one set of load data accurate. If the Manager does not send an update before the DecayWindow expires, the Interceptor will return to the default settings that it loaded on start-up.

Description

Optional: Text String

This optional parameter is an arbitrary string for users that describes this Interceptor instance

KeyFile

Required: Absolute path to keyfile

This is a required parameter. This parameter's value is the name of a file to use as a secret key. The key itself is generated during the installation and must be identical for each of the various WebSpective components. The keyfile is not used in processing HTTP requests.

LocalIP

Required: Interceptor's IP address or DNS listed name

This parameter specifies the IP interface at which the Interceptor listens for connections from the Manager.

LocalPort

Optional: Port number

Default: 4040

The IP port on which the Interceptor listens for connections from the Manager. If this parameter is not set, then the Interceptor will perform as well as the static system information permits.

Log

Optional: Path to desired log file

When specified, the Log parameter provides the Interceptor with the name of a file to which it will log events. If this parameter is not specified, the interceptor will default to the system event log—"syslog" on UNIX systems and the Event Viewer on Windows NT systems.

ManagerIP

Required: Manager's IP address or DNS listed name

This parameter specifies the IP interface at which the Manager listens for connections from other WebSpective components.

ManagerPort

Optional: Port number

Default: 4040

The IP port on which the Manager listens for connections from other WebSpective components.

ReadOnly

Optional: True (1) or False (0)

Default: 0

A true/false parameter. If this parameter is set, the Interceptor will not update its configuration as it runs. This provides better data security to the configuration, but may cause sub-optimal system performance. If the Interceptor's changes at run-time are lost (due to the Interceptor's machine crashing, for instance), then a restarted Interceptor will not receive dynamic configuration changes until a connection is made to the Manager.

RecvTimeout

Optional: Integer

Default: 5

This is an optional parameter which specifies how long to wait, in seconds, for data from a client before giving up and disconnecting. Too low a value will result in clients on slow network connections having difficulty establishing a connection; too high a value may result in many threads waiting for user traffic, and will create a security vulnerability to a denial-of-service attack.

ReusePort

Optional: True (1) or False (0)

Default: 0

This optional parameter controls whether or not the Interceptor will try to use a listening port if it is already in use. If this parameter is set to False (0), the Interceptor will not start if the port is in use. If the parameter is set to True (1), the Interceptor will start up expecting the current processes to die and leave the Interceptor in place.

SecureMessageTimeout

Optional: Integer

Default: 60

The age for which a message to the Interceptor should be accepted from other components in the WebSpective system. This parameter has been established as a defense against "replay" attacks. The value is a number of seconds. When entering a value, consider possible differences in system clock readings and network latency. Otherwise you run the risk of invalidating all of the messages that other components send to the Interceptor.

SSLPassword

Optional: Text string

This parameter contains the Interceptor's secure server password (if an application requires it). Note that the AcceptSSL parameter must be set to true for the Interceptor to handle SSL requests.

UseRefreshPage

Optional: True (1) or False (0)

This setting applies only to requests recieved on the SSL port of a secure application. If an interceptor does not have any secure applications using SSL, then this setting can be ignored.

For applications using SSL this setting controls whether or not a page is sent to the browser when redirecting the SSL request to a server. Older browsers which do not support automatic redirection require a page to be sent with a link for the user to follow into the SSL content.

Most browsers now support automatically following the redirected request, however, if the site wishes to support older browsers then this option should be set to 1. The following browsers do not need this option to be set: Microsoft Internet Explorer 3.0 and later, Netscape 2.0 and later.

Manager Registry

Backup Interval

Optional: Integer

Default: 300

The Manager will backup its object database (ODB) at this interval (in seconds). If the ODB has not changed over the interval, a backup is not made.

Database

Required: Directory path to the Object Database (ODB)

This parameter specifies the location of the Manager's ODB file. This database contains information pertinent to the servers and applications running on every host at a given web site.

Forcesync

Optional: True (1) or False (0)

Default: 0

This parameter dictates the behavior of the Manager when its ODB conflicts with an Agent's ODB. When set to True, the two databases are merged so that each ODB comes away with the same set of information. In cases where a specific datum conflicts, the Manager ODB overwrites the Agent ODB. When set to false, a discrepancy between the databases will cause the Manager to switch into Offline mode for user-governed reconfiguration. See the Online parameter for more information

JDBCDBSpec

Required: Text String

This parameter identifies the external database to the Manager. A typical database spec appears as follows:

```
jdbc:oracle:jdbctest
```

Where "oracle" is the database driver's identifier and "jdbctest" is the system name for the database.

JDBCPassword

Required: Text string

This parameter stores the password the Manager will use to access the LDB (see appendix B for information on the LDB). This is only necessary if an LDB is being used with the system.

JDBCUser

Required: Text string

The user name that the Manager will use to access the LDB (see appendix B for information on the LDB). This is only necessary if an LDB is being used with the system.

KeyFile

Required: Absolute path to keyfile

This is a required parameter. This parameter's value is the name of a file to use as a secret key. The key itself is generated during the installation and must be identical for each of the various WebSpective components. The keyfile is not used in processing HTTP requests.

LocalIP

Required: Interceptor's IP address or DNS listed name

This parameter specifies the IP interface at which the Interceptor listens for connections from the Manager.

LocalPort

Optional: Port number

Default: 4040

The IP port on which the Interceptor listens for connections from the Manager. If this parameter is not set, then the Interceptor will perform as well as the static system information permits.

Name

Required: Text string

The "name" parameter is used to identify the Manager. WebSpective 1.0 only supports one Manager, which is automatically named "manager". In future releases, you will be able to configure additional managers and backup managers. On UNIX systems, the "name" parameter is kept inside the Manager's configuration file. On Windows NT systems it is included in the registry.

Online

Optional: True (1) or False (0)

Default: 1

Reflects the status of the Manager. When the Manager is online, it interacts with other WebSpective components in a normal fashion. When the Manager is offline, it will neither send nor receive information to and from other components. The Manager can be set to offline in situations where the system administrator wants to make a number of changes to the Manager's ODB and have all of the changes sent out to the Agents simultaneously on Restarting.

UserDataDir

Optional: Path to directory

The name of the directory into which user data is about viewer configuration is stored. If you do not specify a path, information is written to the directory the manager was started in.

Viewer Registry

ImageDirect ry

Optional: Absolute path to image directory

This parameter dictates the location to which the Viewer will save image files. Image files contain all of the user preferences and object layout information which have been set during Viewer sessions. If you do not specify a directory, the Viewer will write image files to the directory in which it was started.

K yFil

Required: Absolute path to keyfile

This is a required parameter. This parameter's value is the name of a file to use as a secret key. The key itself is generated during the installation and must be identical for each of the various WebSpective components. The keyfile is not used in processing HTTP requests.

LocalPort

Optional: Port number

Default: 4040

The IP port on which the Interceptor listens for connections from the Manager. If this parameter is not set, then the Interceptor will perform as well as the static system information permits.

ManagerHost

Required: Manager's host name

This parameter specifies the host upon which the Manager resides.

ManagerPort

Optional: Port number

Default: 4040

The IP port on which the Manager listens for connections from other WebSpective components.

Appendix B: Database Structures

Both the Manager and the Agent make use of databases in logging and reporting system information such as events and performance statistics. One of these databases, the Object Database (ODB), is installed with the components and only used inside of the WebSpective system.

The second database, called the Logging Database (LDB) is used to maintain the history of the website and its components. The engine for the LDB is not supplied with WebSpective. The Manager uses an included driver and user-provided account information to connect with this external database and pass along specified data.

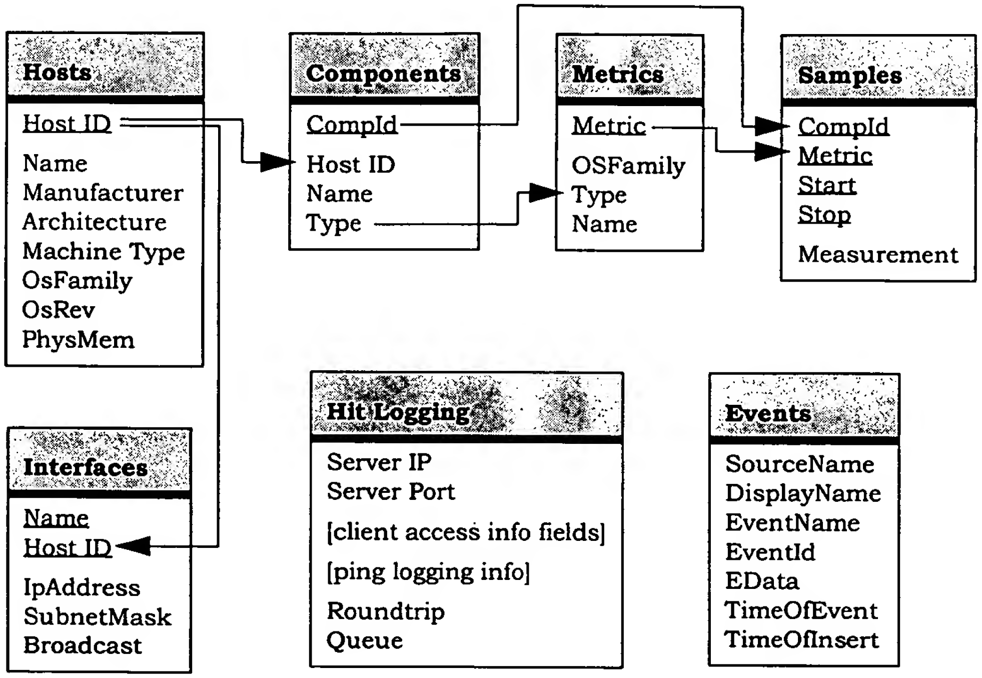
Contents

- Database Architecture
- The Hit Logging Table
- Ping Logging Section
- Metric Logging Tables

Database Architecture

During the Manager's configuration process, it initializes the LDB by creating a number of tables. If you would prefer closer control over the table creation process, please contact customer support about receiving customizable scripts that you can edit and use to manually configure the database.

Here is an Entity Relationship Diagram (ERD) to illustrate how the tables of the LDB relate to each other:



In this diagram, underlined attributes indicate the unique identifiers of their entities. Note that the Hit Logging and Events entities are independent to the other tables. Also note that the primary key in the Samples entity is comprised of all four underlined elements.

Rows in the LDB can reach 4K in size. In an Oracle database, you would configure for this by setting the DB_BLOCK_SIZE parameter to 5K. More details are available in the Oracle manuals under either DB_BLOCK_SIZE or the standard INIT.ORA parameters. Other ODBC databases have similar sizing conventions.

The Hit Logging Table

A subset of the hit logging fields, as specified by the user, is logged for every hit to an application. The subset that is logged will vary from application to application. Note that certain of these fields (identified in italics) will always be gathered from the filter, although the user may choose whether or not to log that data.

TABLE 2. **Hit Logging Table**

Field Name	**Data Type**	**Description**
ServerIP	character(15)	the server endpoint IP, in dotted-decimal notation
ServerPort	integer	the server endpoint port number
ClientIP	character varying (128)	the client endpoint IP, in either DNS or dotted-decimal form
UserName	character (32)	the client username, as provided through the web server
AuthType	character (32)	the type of authentication: "null," "basic," "certified," etc.
Method	character(16)	the HTTP method: GET, PUT, etc.
ClientReq	character varying (1024)	the complete URL requested, including CGI parameters
Protocol	character (16)	the HTTP protocol, e.g. HTTP/1.1
Accept	character varying (128)	the HTTP "Accept" header, indicating the desired data formats to receive
Host	character varying (64)	the HTTP "Host" field, indicating the DNS alias which the user agent used
UserAgent	character (64)	the browser (or robot) making the connection
Connection	character (32)	the HTTP "connection" header
Pragma	character (32)	the HTTP/1.0 "pragma" header, now deprecated
Status	smallint	the HTTP status code
LastMod	character (32)	the last modification date of the content
ContentLen	integer	length of content, in bytes
ContentType	character (32)	the data format of the content
TimeOfHit	character (32)	the moment at which the hit arrived
Duration	real	the number of seconds to process the hit
Referer	character varying (1024)	the HTTP referer field, from whence the user came
Location	character varying (512)	the HTTP location field, to which the user was redirected

TABLE 2. **Hit Logging Table**

Field Name	**Data Type**	**Descripti n**
Aborted	character (5)	Either "true" or "false", to indicate whether a connection was aborted by the browser.
TimeOfInsert	character (32)	The time at which the hit was inserted in the database

Ping Logging Section

The HTTP pings performed by the Agent provide interesting data, in addition to the data reported for user agent requests. This is a part of the Hit Logging table, above. This information is always logged.

Note that even if the server is not responding, ping data has a useful "Status" value indicating the error.

TABLE 3. **Ping Logging Table**

Field Name	**Data Type**	**Description**
RoundTrip	real	The number of milliseconds required for the HTTP transaction, from connect() to close().
Queue	smallint	The estimate of the number of requests waiting in the queue at the time of the ping request. This is calculated by counting the number of hits the Agent processes between the time that it initiates a ping of the web server and the time that the ping is returned. Note that this is an estimate only.

Metric Logging Tables

WebSpective records data about the machines on which it is running. Much of this information is not collected for the hosts themselves, but for pieces of those hosts: each individual disk, each network interface, each CPU. This schema is further complicated by the fact that different operating systems permit us to gather different sets of data about these things, and by the fact that while most data is related to a physical component, some is overall to a machine, and some is measured for processes running on machines (specifically, about the server endpoints).

With that as background, the database schema records the following data about each actual host:

TABLE 4. **Hosts**

Field Name	**Data Type**	**Description**
HostId	character (32), primary key	The hostid or MAC address of the host, as a unique and (nearly) unchanging identifier
Name	character (64)	The hostname of the machine
Manufacturer	character (64)	The maker of the machine
Architecture	character (16)	The manufacturer's architecture specification, usually the chip set used (e.g. NT's x86, Alpha)
MachineType	character (32)	The manufacturer's machine "type" designation
OsFamily	character (16)	The OS family (e.g. WIN32_NT, SunOS)
OsRev	character (32)	The revision of the OS
PhysMem	smallint	Mb of physical RAM in the machine

Within every host, there will be some number of components, about which we record the following information:

TABLE 5. **Components**

Field Name	**Data Type**	**Description**
CompId	integer, primary key	A WebSpective-assigned identifier for this particular component
Host	character (32)	A HostID, from the "Hosts" table, identifying which host holds this component
Name	character (64)	The name of the component
Type	character (32)	A string identifying the type of the component (e.g. "Processor" for CPUs)

Every disk, every CPU, every network interface has a distinct entry here. In addition, there is an entry for every host, of type "System," for overall measurements, and an entry for every endpoint, for computed load information. However, the set of measurements possible for each type of component will vary from operating system to operating system; they are stored in a separate table:

TABLE 6. **Metrics**

Field Name	Data Type	Description
Metric	integer, primary key	A WebSpective-assigned identifier of the available metric.
OSFamily	character (16)	The OsFamily (as in the Hosts table) for which the metric is available.

TABLE 6. **Metrics**

Field Name	Data Type	Description
Type	character (32)	The type of component to which the metric applies
Name	character (32)	The name of the metric, e.g. ("% Time Idle" or "Bytes Read/second")

Finally, periodically each Agent will sample all of those metrics and report them, and periodically the Manager will compute utilization metrics for each endpoint and report those. This, the "actual data" being collected, is recorded like this:

TABLE 7. **Samples**

Field Name	**Data Type**	**Description**
CompId	integer, primary key	The identifier of the component being measured
Metric	integer, primary key	The identifier of the metric being measured
Start	character (32), primary key	The start time of the measurement interval
Stop	character (32), primary key	The stop time of the measurement interval
Value	real, primary key	The measurement value

Interfaces Table

This is used primarily to feed data to the viewer, so that when editing things on a particular host, for example, only sensible network addresses are offered for that host.

TABLE 8. **Interfaces**

Field Name	**Data Type**	**Description**
Name	character (32), primary key	The hardware name of the interface
Host	character (64), primary key	The hostid containing the interface
IpAddress	character (15)	The dotted-decimal IP address
SubnetMask	character (15)	The subnet mask for the IP address
Broadcast	character (15)	The broadcast address for the IP

Events Tabl

The various events tracked in WebSpective are also packaged into a table at the Manager database. This allows the table to be queried on the Viewer's behalf, to provide users with a historical listing of messages.

TABLE 9. **Events**

Field Name	Data Type	Description
SourceName	character (64)	The internal name of the WebSpective entity originating the event
DisplayName	character varying (128)	The user-assigned, familiar name of the originating entity
EventName	character varying (64)	A human-readable name for the event type
EventId	integer	An event code for the event type
EData	character varying (1024)	A string describing the event, with format and contents depending on the particular type of event.
Context	character varying (254)	Contextual information provided with the event
TimeOfEvent	character(32)	The time the event occurred
TimeOfInsert	character(32)	The time the event reached the database
Ack	integer	Field indicating whether or not an event has been acknowledged

Appendix C: System Behavior

The following sections discuss WebSpective's reaction to specified events.

Logged Events

The following events are recorded to the system log.

- Interceptor start
- Interceptor shutdown (with client IP)
- Interceptor minUpdateTime expiration
- Interceptor registration
- Authentication failure
- Timeout
- Create/Delete application
- Deactivate/Reactivate application
- Create/Delete endpoint
- Deactivate/Reactivate endpoint
- Change properties (any of them, at any level, except load updates)

Client Connections

If a request arrives at the endpoint for an application, the Interceptor processes it according to these rules:

- If the application is in state Normal, and there is at least one endpoint for it, then the Interceptor replies with an HTTP "302" redirection request sending the client to one of the application's servers.
- If the application is in state Normal, but there are no servers for it, then the content of the sorry page is returned as content. If a sorry page is not defined, a generic "503" error message saying that no servers are available is returned.
- If the application is Deactiavted, then either the contents of the sorry page is returned, or, if that is not defined, a generic "500" error message saying that the site has been temporarily disabled is returned.

Appendix D: Security Considerations

This appendix is intended to provide you with a basic discussion of WebSpective's security infrastructure. Additionally, you will find information on how to make WebSpective conform to your organization's security policies and standards.

The Keyfile

During installation, you will be asked to locate or generate a keyfile. The keyfile contains a random 160 bit string of code that each component of the WebSpective system uses to verify commands sent by other components. If the components do not each have an identical copy of the keyfile, they will not be able to communicate. In this fashion, intruders cannot mimic the WebSpective system and send faulty commands or information.

The keyfile needs to be generated only once—during the installation of your first WebSpective component. This file is copied to the other components during their own installation. Once created, access to the keyfile should be controlled so that only WebSpective components can read it, and nothing can write to it.

Component Communication

The HTTP ports to which the Interceptor listens are protected against hung or invalid requests and denial-of service attacks (see the CloseDelay property on page 32 and the LingerTimeout property on page 33). Connections between components are authenticated and validated (to prevent spooling attacks) with the KeyFile. None of these connections are encrypted.

Components and *setuid* (UNIX systems only)

Often, WebSpective components will be configured to listen on the default HTTP and HTTPS ports, 80 and 443. To acquire these resources, the components must have administrative permissions. This is accomplished through the use of *setuid* and *setgid,* which allow the components to acquire the ports a root and then demote themselves to user “nobody”.

If you do not want the Interceptor and Agents to use setuid and setgid during their operation, then you have two options. You must either run these process as root or assign them to ports which do not require root permission. Additionally, you must ensure that your web servers are in the same group as the Agent—otherwise, the Agent will not be able to read Filter information from shared memory.

Filesystem security

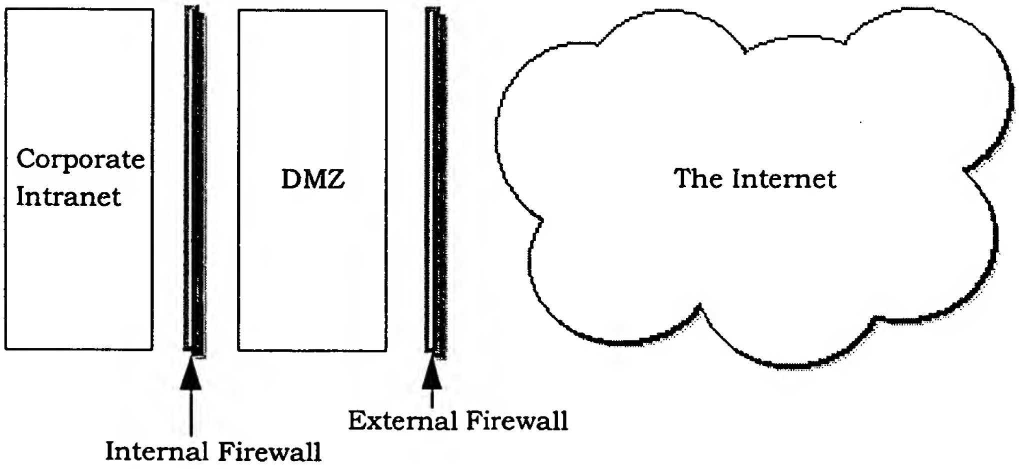
WebSpective configuration files may include sensitive information, such as database passwords and secret keys for authentication. Similarly, the running memory images of the components may have sensitive data within them. It is the site administrator's job to maintain physical security of the host and of its filesystem to be sure these resources are not compromised.

Appendix E: Firewall Considerations

Firewalls are an effective and commonly used security measure at commercial web sites. By design, all traffic coming from the internet or going out from an internal network pass through the firewall, which restricts this traffic in accordance with the security policies of the site.

The WebSpective components communicate with each other via a securely encrypted TCP socket-based protocol. This design choice was made so that they can communicate across a WAN, or even the internet, and across any type of physical network (ethernet, switched ethernet, FDDI, etc.). This means that they must be able to initiate TCP socket connections to each other. Ideally a connection is possible in either direction between any pair of components. However, it is possible to run the system with connections allowed in only one direction.

Most corporations sandwich their website between two firewalls (creating a "DMZ"), isolating it from both the internet and the corporate network.

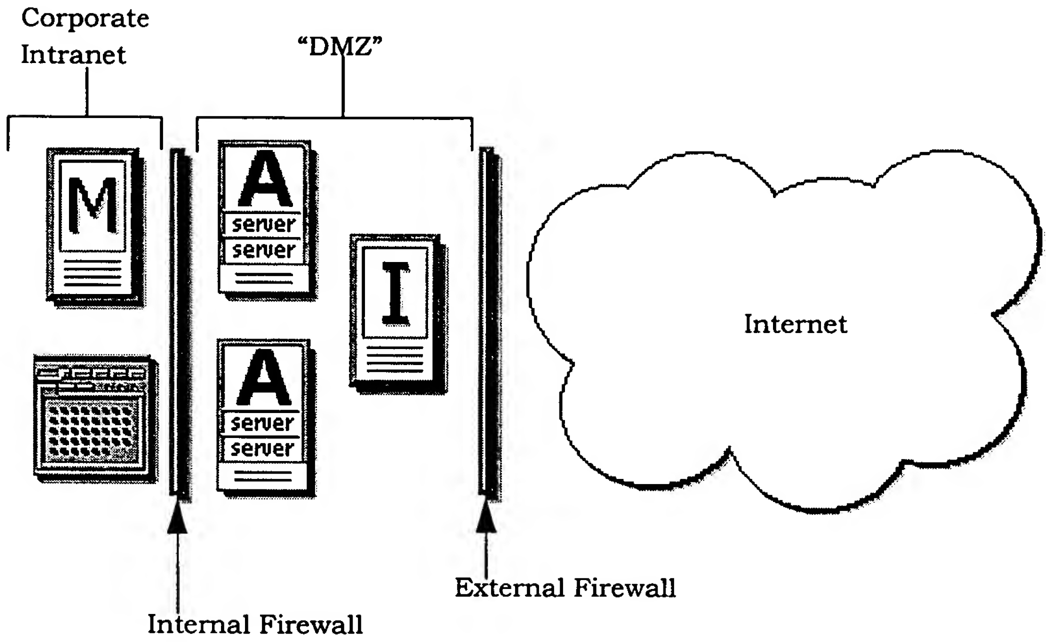


If all of the Webspective components are installed and run within the DMZ, there will be nothing to impede communication. If your environment is simpler than this (with all components running inside or outside of the firewall) then there will also be no issues. However, if individual components are separated by firewalls or packet filters, additional configuration may be required.

How to Incorporat Fir walls

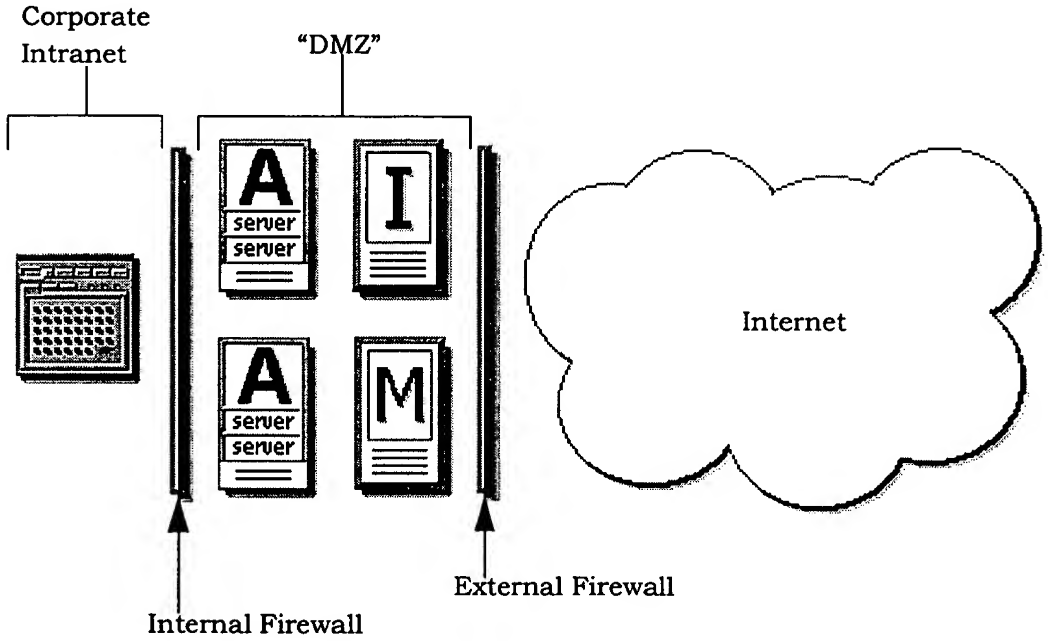
By default the Manager listens at port 5040, and all Agents listen at port 6040 and the Interceptor listens at port 6040. The viewer can be configured to listen for information from the manager at a specific port. We'll assume it is configured to listen at port 5041.

Scenario 1: Suppose you are employing a "DMZ" security approach. All of the Agents and the Interceptor lie within the DMZ. However, the manager and the viewer will be run within the corporate network.



In this case you would probably want to configure the firewall and packet filters between the DMZ and corporate network such that the manager machine is allowed to initiate TCP connections to port 6040 on all the machines on which Agents run, and port 4040 where the Interceptor runs. For optimum communications you also want to allow all of the Agent and Interceptor machines to initiate connections to port 5040 on the manager machine. No configuration changes are required for the Viewer and Manager to talk, since they are both in the corporate intranet.

Scenario 2: Again following a DMZ approach, suppose the Manager is also within the DMZ, and only the viewer will run within the corporate network.



No configuration changes are required for the Agents, Interceptor and Manager to communicate. However, it must be possible for the viewer machine to initiate connections to port 5040 of the Manager machine. For optimum communications you can also allow the Manager machine to initiate connections to port 5041 on the Viewer machine.

For Further Information. If you require a solution which is different from these scenarios or would like further assistance in configuring the WebSpective system around your firewall architecture, please contact Atreve Technical Support at:

support@atreve.com
(617) 576-3400

Glossary

Application. A collection of endpoints which serve related content. An ISP might choose to categorize each client on their web site as an application with its own specific body of content.

Drill Down. To double click on an icon which represents the parent in a hierarchical organization to view its children. An example would be a graphical directory tree in which you could double-click on directory folders to view their contents.

Endpoint. As used by Atreve Software, Inc., an endpoint is an IP address and port number which locate a specific connection to a web server. In the WebSpective management model, each of these endpoints exists to provide the content of a single Application. In general, a given web server will only have one endpoint. However, in the case of a hardware virtual server or other form of "Multi-homing", a single web server can have multiple active ports. The endpoint concept works to differentiate between these different ports.

Firewall. A system to provide site security by restricting the access available. Generally, there are at least two barriers erected; any machine outside the bastion may connect to specific ports of specific machines in the middle zone ("in the firewall", and not at all to machines beyond it ("behind the firewall"). All other requests are blocked. Specific, known machines within the middle zone may connect to specific, known ports of other machines there, or of machines behind the inner barrier. Again, all other requests are blocked. Thus, it should be impossible for an intruder to touch a machine within the wall, and should one of those fall, it should be impossible again for the intruder to widen the breach.

Hardware virtual servers. A form of multi-homing by which a single server process may imitate multiple servers. The server listens simultaneously on several IP interfaces, and the interface of the connection affects the server's behavior. Thus, for Netscape servers (from which the term is borrowed), the content offered varies based on the endpoint.

ISP. *Internet Service Provider.* A company that sells some combination of internet connectivity, disk space, and web server space to its clients.

KeyFil . The generic name of a special file containing a 128 bit key. This file is used by components of the WebSpective system to authenticate connections made between them.

LDB. *Logging Database.* This is an external database used by WebSpective to keep and correlate historical web site data. See appendix B for more information.

Multi-homing. A method of offering multiple servers on a single machine. This might be by hardware virtual servers, or by software virtual servers, or it might be simply by having multiple server processes on the same host.

ODBC. Open Database Connectivity^{TM}. A protocol for interaction with a database.

Software virtual servers. A form of multi-homing in which a single NetScape server process, with a single listening endpoint, changes its behavior based on the DNS name on which the connection was made. This is impossible to detect from the server's perspective, and relies on the use of the optional “Host” header field in the HTTP request.

Sorry Page. A web page that is posted when applications are unavailable. In WebSpective usage, "sorry page" may also refer to the URL which points to the page itself.

Index

A

Accepted MIME types 39
Agent
  defined 3
AppIPAddr 31
AppPort 31

B

Browser IP 38

C

Close Delay 32
Command TTL 32
Comment 32
Content length 39
Content modification date 39
Content type 39
Creation Bar 23
custom views 22

D

Deactivated 29
Deactivated URL 32
DecayWindow 32, A-5
Duration of hit 39

F

Failed 29
Filter
  defined 4
  installation 13

H

hardware
  property 32
  requirements 4
hardware virtual server 8
Health Tab 23
host
  property 32
Host ID 32
Host Name 33
How to
  add items to the logging fields 39
  add objects 40
  change a property globally 31
  create a new view 22
  deactivate an object 41
  delete a view 23
  delete an object 43
  edit a custom view 23
  reactivate an object 42
  restart an object 42
  save a view under a new name 23
  stop an object 42
HTTP AuthDB 38
HTTP AuthType 38
HTTP Connection 39
HTTP Cookie(s) 39
HTTP Host field 39
HTTP Pragma 39
HTTP Status 39

I

installation
  basics 6
  on UNIX systems 11
  on Windows systems 9
Interceptor
  defined 2
IP/name 33
IPCchannel 33
IsService 33

K

KeyFile
  defined D-1
  property 33

L

LDB B-1
  architecture B-1
  Events Table B-7
  Hit Logging Table B-3
  Interfaces Table B-6
  Metric Logging Tables B-4
    Components B-5
    Hosts B-5
    Metrics B-5
    Samples B-6
  Ping Logging Table B-4
LingerTimeout 33
Logging Fields
  Accepted MIME types 39
  Browser IP 38
  Content length 39
  Content modification date 39
  Content type 39
  Duration of hit 39
  HTTP AuthDB 38
  HTTP AuthType 38
  HTTP Connection 39

  - HTTP Cookie(s) 39
  - HTTP Host field 39
  - HTTP Pragma 39
  - HTTP Status 39
  - property 33
  - Redirection location 39
  - Referer URL 39
  - Requested URL 38
  - Server file for URL 38
  - setting 38
  - Time of hit 39
  - User Aborted? 39
  - User agent 38
  - User name 38

M

- Management Tab 22
- Manager
  - defined 3
- Maxed Out 29
- Mgmt IP/name 34
- Mgmt Log 34
- Mgmt Port 34
- MgtInterval 34
- multi-IP web server 7
- multi-process server 7

N

- Name 34
- Normal 29

O

- object
  - management 40
  - properties 30
    - AppIPAddr 31
    - AppPort 31
    - CloseDelay 32
    - Command TTL 32
    - Comment 32
    - Deactivated URL 32
    - DecayWindow 32, A-5
    - Editor 30
    - Hardware 32
    - Host 32
    - Host ID 32
    - Host Name 33
    - IP/name 33
    - IPCchannel 33
    - IsService 33
    - KeyFile 33
    - LingerTimeout 33
    - listings 31
    - Logging Fields 33
    - Mgmt IP/name 34
    - Mgmt Log 34
    - Mgmt Port 34
    - MgtInterval 34
    - Name 34
    - Operating System Revision 34
    - Physical Memory 35
    - PingTimeout 35
    - Platform 35
    - PortTakeover 35
    - RecvTimeout 35
    - RedirectURL 35
    - Registry 35
    - ReviveInterval 36
    - ReviveRetries 36
    - SecureServer 36
    - ServiceName 36
    - SetupTime 36
    - ShutdownURL 36
    - SSLPassword 36
    - StartDir 37
    - Startup CmdLine 37
    - State 37
    - Strength 37
    - Threads 37
    - TimeRevived 37
    - Transaction List 38
    - TransTimeout 38
  - states 28
    - Deactivated 29
    - Failed 29
    - Maxed Out 29
    - Normal 29
    - Off 30
    - Problem 29
    - Restarting 29
    - Starting 28
    - Stopping 28
    - Unknown 29
- Off 30
- Operating System Revision 34

P

- permissions D-1
- Physical Memory 35
- PingTimeout 35
- Platform 35
- PortTakeover 35
- Problem 29

R

- RecvTimeout 35
- RedicrectURL 35
- Redirection location 39
- Referer URL 39
- Registry 35
- Requested URL 38
- Restarting 29
- ReviveInterval 36
- ReviveRetries 36

S

- SecureServer 36
- Security 5
- Server file for URL 38
- ServiceName 36

SetupTime 36
ShutdownURL 36
software virtual server 8
SSLPassword 36
StartDir 37
Starting 28
Startup CmdLine 37
State 37
Stopping 28
Strength 37

T

Tables
  Events B-7
  Hit Logging B-3
  Interfaces B-6
  Metric Logging B-4
    Components B-5
    Hosts B-5
    Metrics B-5
    Samples B-6
  Ping Logging B-4
Threads 37
Time of hit 39
TimeRevived 37
Transaction List 38
transaction list 41
TransTimeout 38

U

Unknown 29
User Aborted? 39
User agent 38
User name 38

V

Viewer
  Creation Bar 23
  custom views 22
  defined 4
  Health Tab 23
  interface 21
  Management Tab 22
  navigation 21

W

WebSpective
  defined 1